Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE

Corso di Laurea in Matematica

ESPONENZIALE DI MATRICE: ASPETTI COMPUTAZIONALI

Tesi di Laurea in Calcolo Numerico

Relatore:
Chiar.ma Prof.ssa
VALERIA SIMONCINI

Presentata da:
ANDREA RAMI

Sessione Unica
Anno Accademico 2016-2017

Introduzione

Le funzioni di matrici giocano un ruolo importante in molte applicazioni. Tra le funzioni di matrici più studiate in analisi numerica vi è l'esponenziale di matrice che è fondamentale nell'ambito delle equazioni differenziali lineari. Infatti un problema vettoriale del tipo

$$\frac{dy}{dt} = At, \qquad y(0) = c, \qquad y \in \mathbb{C}^n,\ A \in \mathbb{C}^{n\times n},$$

ha come soluzione $y(t) = e^{At}c$. Più in generale, supponendo adeguatamente f liscia, la soluzione del sistema non omogeneo

$$\frac{dy}{dt} = Ay + f(t,y), \qquad y(0) = c, \qquad y \in \mathbb{C}^n,\ A \in \mathbb{C}^{n\times n}$$

soddisfa

$$y(t) = e^{At}c + \int_0^t e^{A(t-s)} f(s,y)ds, \tag{1}$$

che è una formula esplicita per y nel caso in cui f sia indipendente da y.

Analogamente, alcune equazioni differenziali di matrici hanno soluzioni che si possono esprimere in termini di esponenziali di matrice. Ad esempio, si può verificare facilmente che la soluzione di

$$\frac{dY}{dt} = AY + YB, \qquad Y(0) = C$$

è

$$Y(t) = e^{At}Ce^{Bt}.$$

Tra le diverse tecniche utilizzate per calcolare la soluzione numerica delle equazioni differenziali ordinarie, vi sono alcuni metodi che usano esplicitamente l'esponenziale di matrice.

Viene fatto uso dell'esponenziale di matrice in diverse applicazioni. Tra queste vi è la spettroscopia di risonanza magnetica nucleare bidimensionale (NMR), che è uno

strumento per determinare la struttura e le dinamiche delle molecole nella soluzione, i modelli di Markov e la teoria dei controlli.

Vista l'importanza ricoperta dall'esponenziale di matrice, in questo testo verranno mostrate alcune sue proprietà, seguite da un importante metodo di calcolo e da altri algoritmi per il calcolo dell'esponenziale di matrice, mostrando con un esempio numerico le differenze tra i diversi metodi.

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Proprietà di base e condizionamento

L'importanza dell'esponenziale di matrice deriva dal suo ruolo chiave nella risoluzione delle equazioni differenziali dove, a seconda dell'applicazione, il problema può essere quello di calcolare e^A data A, di calcolare e^{At} dati A e molti t, o di applicare e^A o e^{At} ad un vettore. Se ne dà prima definizione.

Definizione 1.1. Data una matrice $A \in \mathbb{C}^{n\times n}$, si definisce l'*esponenziale di matrice* come

$$e^A = \sum_{i=0}^{\infty} \frac{A^i}{i!} = I + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \cdots .$$

Un'esponenziale di matrice viene quindi approssimata da una serie di Taylor con raggio di convergenza infinito.

Un'altra rappresentazione di un'esponenziale di matrice è

$$e^A = \lim_{s\to\infty} \left(I + \frac{A}{s} \right)^s$$

in cui e^A viene vista come limite del primo ordine dello sviluppo di Taylor di $\left(\frac{A}{s}\right)^s$, con $s \in \mathbb{Z}$. In generale, si può prendere il limite per $r \to \infty$ o $s \to \infty$ di r termini dello sviluppo di Taylor di A/s elevato alla s che generalizzano entrambe le due rappresentazioni precedenti.

Nel seguito si farà uso delle seguenti definizioni.

Definizione 1.2. Una *norma* su $\mathbb{C}^{m\times n}$ è una funzione $\|\cdot\| : \mathbb{C}^{m\times n} \to \mathbb{R}^+$ tale che per ogni matrice A, B e ogni scalare λ si ha che

1. $\|A\| = 0$ se e solo se $A = 0$ (matrice nulla)
2. $\|\lambda A\| = |\lambda|\|A\|$
3. $\|A + B\| \leq \|A\| + \|B\|$.

In particolare, se $m = n$ si ha anche che $\|AB\| \leq \|A\|\|B\|$.

Definizione 1.3. Si definiscono alcune particolari norme di matrici:

- **norma di Frobenius**: $\|A\|_F = \left(\sum_{i,j} |a_{ij}|^2\right)^{\frac{1}{2}}$
- **norma 2 indotta**: $\|A\|_2 = \max_{0 \neq x \in \mathbb{R}^n} \frac{\|Ax\|_2}{\|x\|_2}$, dove $\|x\|_2 := \left(\sum_{i=1}^{n} |x_i|^2\right)^{\frac{1}{2}}$
- **norma p indotta**: $\|A\|_p = \max_{0 \neq x \in \mathbb{R}^n} \frac{\|Ax\|_p}{\|x\|_p}$, dove $\|x\|_p := \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}}$
- **norma 1 indotta**: $\|A\|_1 = \max_{0 \neq x \in \mathbb{R}^n} \frac{\|Ax\|_1}{\|x\|_1}$, dove $\|x\|_1 := \left(\sum_{i=1}^{n} |x_i|\right)$
- **norma ∞ indotta**: $\|A\|_\infty = \max_{i,j} |a_{ij}|$

Definizione 1.4. Una norma matriciale si dice *compatibile* se $\|AB\| \leq \|A\|\|B\|$ per ogni $A \in \mathbb{C}^{m \times n}$ e $B \in \mathbb{C}^{n \times p}$. In particolare, la norma di Frobenius e tutte le norme indotte sono compatibili.

Un importante teorema fornisce una formula generale per ottenere e^A, dando anche un errore legato a r e s finiti. Prima di enunciarlo però, si mostra un risultato che verrà sfruttato nella dimostrazione del teorema.

Teorema 1.1 (Errore di troncamento delle serie di Taylor)**.** *Sia f una funzione e si supponga che abbia sviluppo in serie di Taylor*

$$f(z) = \sum_{k=0}^{\infty} a_k (z - \alpha)^k \quad \textit{con } a_k = \frac{f^{(k)}(\alpha)}{k!}$$

con raggio di convergenza r. Se $A \in \mathbb{C}^{n\times n}$ con $\rho(A-\alpha I) < r$, allora per ogni norma matriciale

$$\left\| f(A) - \sum_{k=0}^{s-1} a_k (A-\alpha I)^k \right\| \leq \frac{1}{s!} \max_{0\leq t\leq 1} \left\| (A-\alpha I)^s f^{(s)}(\alpha I + t(A-\alpha I)) \right\|.$$

Ora si può enunciare e dimostrare il seguente teorema.

Teorema 1.2 (Teorema di Suzuki)**.** *Per $A \in \mathbb{C}^{n\times n}$, sia*

$$T_{r,s} = \left[\sum_{i=0}^{r} \frac{1}{i!} \left(\frac{A}{s} \right)^i \right]^s.$$

Allora per ogni norma matriciale compatibile

$$\|e^A - T_{r,s}\| \leq \frac{\|A\|^{r+1}}{s^r (r+1)!} e^{\|A\|}$$

e $\lim_{r\to\infty} T_{r,s}(A) = \lim_{s\to\infty} T_{r,s}(A) = e^A$.

Dimostrazione. Sia $T = \sum_{i=0}^{r} \frac{1}{i!} \left(\frac{A}{s} \right)^i$ e $B = e^{A/s}$. Allora, poiché B e T commutano,

$$e^A - T_{r,s} = B^s - T^s = (B-T)(B^{s-1} + B^{s-2}T + \cdots + T^{s-1}).$$

Quindi

$$\|e^A - T_{r,s}\| \leq \|B-T\| s \max_{i=0:s-1} \|B\|^i \|T\|^{s-i-1}.$$

Per costruzione di T e per definizione di esponenziale di matrice, si ha che $\|T\| \leq \sum_{i=0}^{r} \frac{1}{i!} \left(\frac{\|A\|}{s} \right)^i \leq e^{\|A\|/s}$. Analogamente, $\|B\|$ soddisfa la stessa relazione. Quindi

$$\|e^A - T_{r,s}\| \leq s \|e^{A/s} - T\| e^{\frac{s-1}{s}\|A\|}.$$

Per il Teorema 1.1, risulta

$$\|e^{A/s} - T\| \leq \frac{1}{(r+1)!} \left(\frac{\|A\|}{s} \right)^{r+1} e^{\|A\|/s}.$$

La tesi segue direttamente da queste due relazioni e le verifiche dei limiti sono immediate.

□

Anche se in generale $e^{A+B} \neq e^A e^B$, l'uguaglianza vale quando A e B commutano, come si può vedere dal seguente risultato.

Teorema 1.3. *Per A, $B \in \mathbb{C}^{n\times n}$, $e^{(A+B)t} = e^{At}e^{Bt}$ $\forall t$ se e solo se $AB = BA$.*

Dimostrazione. Se $AB = BA$ allora tutti i termini nello sviluppo in serie di potenze di $e^{(A+B)t}$ e di $e^{At}e^{Bt}$ commutano e quindi queste matrici sono uguali per gli stessi motivi del caso scalare. Se $e^{(A+B)t} = e^{At}e^{Bt}$ $\forall t$ allora uguagliando i coefficienti di t^2 negli sviluppi in serie di potenze di entrambi i membri si ottiene $(AB+BA)/2 = AB$ o $AB = BA$. □

1.1 Condizionamento

In questa sezione vengono mostrati un paio di risultati riguardanti l'esponenziale di matrice che risulteranno importanti per discutere le conclusioni del capitolo successivo. Si consideri quindi la seguente definizione.

Definizione 1.5. Si chiama *derivata di Fréchet* di una funzione di matrice $f : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ in $X \in \mathbb{C}^{n\times n}$ la mappa lineare

$$L : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$$

$$E \mapsto L(X, E)$$

tale che per ogni $E \in \mathbb{C}^{n\times n}$

$$f(X + E) - f(X) - L(X, E) = o(\|E\|). \tag{1.1}$$

L'esponenziale di matrice soddisfa la seguente identità

$$e^{(A+E)t} = e^{At} + \int_0^t e^{A(t-s)} E e^{(A+E)s} ds,$$

che si ricava vedendo la matrice $e^{(A+E)t}$ come soluzione del tipo (1) di un'equazione differenziale. Usando questa espressione e sostituendo dentro l'integrale si ottiene

$$e^{(A+E)t} = e^{At} + \int_0^t e^{A(t-s)} E e^{As} ds + \mathcal{O}(\|E\|^2).$$

Di conseguenza, da (1.1), si può definire la derivata di Fréchet di una matrice A nella direzione E come segue.

Definizione 1.6. Si definisce la *derivata di Fréchet dell'esponenziale di matrice* come

$$L(A,E) = \int_0^1 e^{A(1-s)} E e^{As} ds, \tag{1.2}$$

dove $E \in \mathbb{C}^{n\times n}$, $E \neq 0$.

In particolare, $L(A)$ è definito dalla seguente relazione

$$\|L(A)\| = \max_{E\neq 0} \frac{\|L(A,E)\|}{\|E\|}.$$

Sfruttando queste definizioni, si può enunciare un lemma che fornisce una stima per il numero di condizionamento relativo

$$\kappa_{\exp}(A) = \frac{\|L(A)\|\|A\|}{\|e^A\|} \tag{1.3}$$

dell'esponenziale.

Lemma 1.4. *Per $A \in \mathbb{C}^{n\times n}$, per ogni norma matriciale subordinata vale*

$$\|A\| \leq \kappa_{exp}(A) \leq \frac{e^{\|A\|}\|A\|}{\|e^A\|}. \tag{1.4}$$

Dimostrazione. Per la definizione (1.2), si ha

$$\|L(A,E)\| \leq \|E\| \int_0^1 e^{\|A\|(1-s)} e^{\|A\|s} ds = \|E\| \int_0^1 e^{\|A\|} ds = \|E\| e^{\|A\|},$$

cosicché $\|L(A)\| \leq e^{\|A\|}$. Inoltre, $\|L(A)\| \geq \|L(A,I)\| = \|\int_0^1 e^A ds\| = \|e^A\|$. La tesi segue facilmente da queste relazioni. □

Un'importante classe di matrici che hanno il numero di condizionamento minore possibile sono le matrici normali.

Definizione 1.7. Una matrice $A \in \mathbb{C}^{n\times n}$ si dice normale se $A^H A = AA^H$, dove A^H indica la matrice trasposta coniugata.

Teorema 1.5 (Teorema di Van Loan)**.** *Se $A \in \mathbb{C}^{n\times n}$ è normale, allora, rispetto alla norma 2, $\kappa_{exp}(A) = \|A\|_2$.*

Dimostrazione. È necessaria una piccola variazione della stima superiore in (1.4). Per prima cosa, si noti che per una matrice normale B, $\|e^B\|_2 = e^{\alpha(B)}$, dove $\alpha(B)$ è il raggio spettrale, definito come $\alpha(B) = \max\{Re(\lambda_i) : \lambda_i \in \Lambda(B)$, $\Lambda(B)$ insieme degli autovalori di $B\}$. Da (1.2), si ha

$$\begin{aligned}\|L(A,E)\|_2 &\leq \|E\|_2 \int_0^1 e^{\alpha(A)(1-s)} e^{\alpha(A)s} ds \\ &= \|E\|_2 \int_0^1 e^{\alpha(A)} ds = e^{\alpha(A)} \|E\|_2 = \|e^A\|_2 \|E\|_2.\end{aligned}$$

Di conseguenza $\|L(A)\|_2 \leq \|e^A\|_2$, il che implica il risultato visto nella stima inferiore in (1.4). □

Capitolo 2

Metodo di scalatura e quadratura

Il metodo di scalatura e quadratura (in inglese "scaling e squaring method") è il metodo più utilizzato per calcolare l'esponenziale di matrice ed è anche il metodo implementato in MATLAB dalla funzione `expm`.
Questo metodo sfrutta la relazione $e^A = (e^{A/\sigma})^\sigma$, per $A \in \mathbb{C}^{n\times n}$ e $\sigma \in \mathbb{C}$, insieme al fatto che e^A può essere approssimato da un approssimante di Taylor o di Padé vicino all'origine, cioè per $\|A\|$ piccole. L'idea è di scegliere σ come una potenza intera di 2 ($\sigma = 2^s$), così che A/s abbia norma di ordine 1; approssimare $e^{A/2^s} \approx r(A/2^s)$, dove r è un approssimante dell'esponenziale di Taylor o di Padé; e quindi prendere $e^A \approx r(A/2^s)^{2^s}$, dove l'approssimazione è ottenuta da s elevamenti al quadrato.

In breve, il metodo scala la matrice A di una potenza di 2 così che abbia norma di ordine 1, calcola un approssimante di Padé dell'esponenziale di matrice ed infine eleva ripetutamente al quadrato per annullare gli effetti della scalatura.

Definizione 2.1. Si indica con $R_{k,m}$ lo spazio delle funzioni razionali con numeratore e denominatore di grado minore o uguale, rispettivamente, a k e m.

In questo contesto si preferisce usare gli approssimanti di Padé rispetto alle approssimazioni delle serie di Taylor in quanto forniscono una precisione fissata con un costo computazionale più basso. Nel seguito viene quindi data la definizione di approssimante di Padé di indici $[k/m]$.

Definizione 2.2. Sia $f(x)$ una funzione scalare. La funzione razionale

$$r_{km}(x) = p_{km}(x)/q_{km}(x)$$

viene chiamata *approssimante di Padé di indici [k/m] di f* se $r_{km} \in R_{k,m}$, $q_{km}(0) = 1$, e

$$f(x) - r_{km}(x) = \mathcal{O}(x^{k+m+1}).$$

In particolare, l'approssimante di Padé di una funzione esponenziale è tale che $e^x - r_{km}(x) = \mathcal{O}(x^{k+m+1})$.
Per ogni k e m, questi approssimanti di Padé sono noti esplicitamente:

$$p_{km}(x) = \sum_{j=0}^{k} \frac{(k+m-j)!k!}{(k+m)!(k-j)!}\frac{x^j}{j!}, \quad q_{km}(x) = \sum_{j=0}^{m} \frac{(k+m-j)!m!}{(k+m)!(m-j)!}\frac{(-x)^j}{j!}.$$

Si noti che $p_{km}(x) = q_{mk}(-x)$, che rispecchia la proprietà della funzione esponenziale $1/e^x = e^{-x}$.
Si vede che r_{km} soddisfa la definizione di approssimante di Padé dall'espressione dell'errore

$$e^x - r_{km}(x) = (-1)^m \frac{k!m!}{(k+m)!(k+m+1)!}x^{k+m+1} + \mathcal{O}(x^{k+m+2}). \tag{2.1}$$

È nota anche l'esatta espressione dell'errore per una matrice $A \in \mathbb{C}^{n\times n}$:

$$e^A - r_{km}(A) = \frac{(-1)^m}{(k+m)!}A^{k+m+1}q_{km}(A)^{-1}\int_0^1 e^{tA}(1-t)^k t^m dt. \tag{2.2}$$

In genere, vengono usati approssimanti diagonali (ovvero approssimanti con $k = m$), in quanto r_{km} con $k \neq m$ è meno preciso di r_{jj}, con $j = \max(k, m)$, e r_{jj} può essere stimato con una matrice con ugual costo. Inoltre gli approssimanti diagonali hanno una proprietà particolare: se gli autovalori di A giacciono sul semipiano aperto sinistro allora gli autovalori di $r_{mm}(A)$ hanno modulo minore di 1. Questa proprietà risulta importante nelle applicazioni alle equazioni differenziali. Per semplificare la notazione, gli approssimanti diagonali verranno indicati con $r_m(x) = p_m(x)/q_m(x)$.

Lo scopo è di scegliere s, la scalatura iniziale $A \leftarrow A/2^s$ e il grado m dell'approssimante di Padé, cosicché l'esponenziale venga calcolata con un errore all'indietro limitato dalla precisione di macchina e con costi computazionali minimi. Nel fare ciò, si suppone di lavorare in aritmetica esatta e di esaminare solamente gli effetti degli errori di approssimazione negli approssimanti di Padé.

La scelta di s dipenderà da $\|A\|$, dove la norma può essere una qualsiasi norma matriciale compatibile. Di conseguenza il primo obiettivo è quello di stimare l'errore all'indietro in termini di $\|2^{-s}A\|$ e quindi determinare, per ogni grado m, il massimo di $\|2^{-s}A\|$ per il quale si può ottenere da r_m l'errore all'indietro desiderato. Sia

$$e^{-A}r_m(A) = I + G = e^H, \tag{2.3}$$

dove G è una matrice con $\|G\| < 1$ tale che esista $H = \log(I+G)$ (dove log denota il logaritmo naturale). Di conseguenza, poiché $\log(I+G) = \sum_{j=1}^{\infty}(-1)^{j+1}\frac{G^j}{j}$, si ha

$$\|H\| = \|\log(I+G)\| \leq \sum_{j=1}^{\infty}\frac{\|G\|^j}{j} = -\log(1-\|G\|).$$

Per costruzione, G è una funzione di A e quindi anche H, che commuta con A. Segue che

$$r_m(A) = e^A e^H = e^{A+H}.$$

Sostituendo $A/2^s$ ad A, con s intero non negativo, ed elevando entrambi i membri dell'equazione alla 2^s, si ottiene

$$r_m(A/2^s)^{2^s} = e^{A+E},$$

dove $E = 2^s H$ soddisfa

$$\|E\| \leq -2^s \log(1-\|G\|)$$

e G soddisfa (2.3) con $2^{-s}A$ al posto di A.
Il seguente teorema riassume quanto detto.

Teorema 2.1. *Sia r_m l'approssimante diagonale di Padé che soddisfa la seguente relazione*

$$e^{-2^{-s}A}r_m(2^{-s}A) = I + G, \tag{2.4}$$

dove $\|G\| < 1$ e la norma è una qualsiasi norma matriciale compatibile. Allora

$$r_m(2^{-s}A)^{2^s} = e^{A+E},$$

dove E commuta con A e

$$\frac{\|E\|}{\|A\|} \leq \frac{-\log(1-\|G\|)}{\|2^{-s}A\|}. \tag{2.5}$$

Questo teorema è un risultato dell'errore all'indietro, in quanto interpreta gli errori di troncamento nell'approssimante di Padé come una perturbazione nella matrice A iniziale. Questo risultato, in particolare, vale per qualsiasi approssimazione razionale r_m, in quanto non sono state usate proprietà specifiche dell'approssimante di Padé. Il vantaggio è che l'errore tiene in considerazione l'effetto della fase di quadratura sull'errore dell'approssimante di Padé e, confrontato con una stima dell'errore in avanti, si evita la necessità di considerare il condizionamento del problema.

Lo scopo ora è di stimare la norma di G in (2.4) in termini di $\|2^{-s}A\|$. Si può procedere in due modi. Il primo consiste nell'assumere una stima dall'alto di $\|A\|$ e usare la formula dell'errore (2.2) per ottenere una stima esplicita di G, o almeno una che sia facile da valutare. Si riscrive quindi la formula (2.2) come

$$G = e^{-A} r_{km}(A) - I = \frac{(-1)^{m+1}}{(k+m)!} A^{k+m+1} q_{km}(A)^{-1} \int_0^1 e^{(t-1)A}(1-t)^k t^m dt.$$

Passando alle norme, si ha

$$\begin{aligned} \|G\| &\leq \frac{\|A\|^{k+m+1}}{(k+m)!} \|q_{km}(A)^{-1}\| \int_0^1 e^{(1-t)\|A\|}(1-t)^k t^m dt \\ &= \frac{\|A\|^{k+m+1}}{(k+m)!} \|q_{km}(A)^{-1}\| \int_0^1 e^{t\|A\|} t^k (1-t)^m dt. \end{aligned}$$

Sostituendo in (2.2) $\|A\|$ ad A, la stima può essere riscritta come

$$\|G\| \leq q_{mk}(\|A\|) \|q_{km}(A)^{-1}\| |e^{\|A\|} - r_{mk}(\|A\|)|.$$

Ora, prendendo $k = m$ e sfruttando le proprietà di $q_m(A)$ per riscrivere la stima in modo che dipenda solo da $\|A\|$, si ottiene

$$\|G\| \leq \frac{q_m(\|A\|)}{2 - q_m(-\|A\|)} \left| e^{\|A\|} - r_m(\|A\|) \right|$$

con $\|G\| < 1$ e $q_m(-\|A\|) < 2$.
Questo approccio, sebbene sia matematicamente elegante, non fornisce la miglior stima possibile e, quindi, il miglior algoritmo. Di conseguenza si userà una stima della norma di G che non preveda supposizioni a priori sulla norma di A. Seguendo questo approccio, la stima è difficile da valutare; è, tuttavia, solo un piccolo inconveniente, in quanto vi è la necessità di fare la stima solo durante la progettazione dell'algoritmo.

Si definisca la funzione $\rho(x) = e^{-x} r_m(x) - 1$. Considerando la proprietà (2.1) dell'approssimazione di Padé, ρ ha uno sviluppo in serie di potenze

$$\rho(x) = \sum_{i=2m+1}^{\infty} c_i x^i \tag{2.6}$$

che converge assolutamente per $|x| < \min\{|t| : q_m(t) = 0\} =: \nu_m$. Quindi

$$\|G\| = \|\rho(2^{-s}A)\| \leq \sum_{i=2m+1}^{\infty} |c_i|\theta^i =: f(\theta), \tag{2.7}$$

dove $\theta := \|2^{-s}A\| < \nu_m$. Se A è una matrice qualsiasi ed è noto solo il valore di $\|A\|$ allora (2.7) fornisce la più piccola stima possibile di $\|G\|$.

Combinando (2.7) con (2.5), si ottiene

$$\frac{\|E\|}{\|A\|} \leq \frac{-\log(1 - f(\theta))}{\theta}. \tag{2.8}$$

Calcolare $f(\theta)$ in (2.7) sarebbe facile se i coefficienti c_i fossero concordi: si avrebbe $f(\theta) = |\rho(\theta)|$. Sperimentalmente, i c_i sono concordi per qualche m, ma non tutti. Usando il Symbolic Math Toolbox di MATLAB, si calcola $f(\theta)$, e quindi la stima (2.8), in 250 cifre digitali in aritmetica finita, raccogliendo i primi 150 termini della serie, dove i c_i in (2.6) sono ottenuti simbolicamente. Per $m = 1 : 21$ si cercano gli zeri per determinare il più grande valore di θ, indicato con θ_m, tale che la stima (2.8) dell'errore all'indietro non sia maggiore di $u = 2^{-53} \approx 1.1 \times 10^{-16}$. I risultati sono mostrati nella Tabella 2.1.

Nella seconda riga della tabella, vengono mostrati i valori di ν_m e si vede che $\theta_m < \nu_m$ sempre, confermando quindi la validità della stima (2.7). La disuguaglianza $\theta_m < \nu_m$ conferma anche il fatto che $q_m(A)$ è non singolare (quindi con determinante diverso da 0) per $\|A\| \leq \theta_m$.

Si deve ora determinare il costo del calcolo di $r_m(A)$. Come accennato precedentemente, esiste una relazione che lega i polinomi a numeratore e a denominatore dell'approssimante di Padé: $q_m(x) = p_m(-x)$. Di conseguenza, si può basare uno schema efficiente sul calcolo esplicito delle potenze pari di A, costruendo p_m e q_m e risolvendo l'equazione matriciale $q_m r_m = p_m$. Se $p_m(x) = \sum_{i=0}^{m} b_i x^i$, per i gradi pari,

$$\begin{aligned} p_{2m}(A) &= b_{2m}A^{2m} + \cdots + b_2 A^2 + b_0 I + A(b_{2m-1}A^{2m-2} + \cdots + b_3 A^2 + b_1 I) \\ &=: U + V, \end{aligned}$$

Tabella 2.1: Valori massimi di θ_m di $\|2^{-s}A\|$ tali che la stima dell'errore all'indietro (2.8) non sia maggiore di $u = 2^{-53}$, valori di $\nu_m = \min\{|x| : q_m(x) = 0\}$, e stima dall'alto ξ_m di $\|q_m(A)^{-1}\|$.

m	1	2	3	4	5	6	7
θ_m	$3.7e-8$	$5.3e-4$	$1.5e-2$	$8.5e-2$	$2.5e-1$	$5.4e-1$	$9.5e-1$
ν_m	$2.0e0$	$3.5e0$	$4.6e0$	$6.0e0$	$7.3e0$	$8.7e0$	$9.9e0$
ξ_m	$1.0e0$	$1.0e0$	$1.0e0$	$1.0e0$	$1.1e0$	$1.3e0$	$1.6e0$

m	8	9	10	11	12	13	14
θ_m	$1.5e0$	$2.1e0$	$2.8e0$	$3.6e0$	$4.5e0$	$5.4e0$	$6.3e0$
ν_m	$1.1e1$	$1.3e1$	$1.4e1$	$1.5e1$	$1.7e1$	$1.8e1$	$1.9e1$
ξ_m	$2.1e0$	$3.0e0$	$4.3e0$	$6.6e0$	$1.0e1$	$1.7e1$	$3.0e1$

m	15	16	17	18	19	20	21
θ_m	$7.3e0$	$8.4e0$	$9.4e0$	$1.1e1$	$1.2e1$	$1.3e1$	$1.4e1$
ν_m	$2.1e1$	$2.2e1$	$2.3e1$	$2.5e1$	$2.6e1$	$2.7e1$	$2.8e1$
ξ_m	$5.3e1$	$9.8e1$	$1.9e2$	$3.8e2$	$8.3e2$	$2.0e3$	$6.2e3$

che può essere stimato con $m+1$ moltiplicazioni tra matrici[1] costruendo A^2, A^4,$\cdots$, A^{2m}. Allora si ottiene

$$q_{2m}(A) = U - V$$

senza costi aggiuntivi. Per gradi dispari,

$$\begin{aligned} p_{2m+1}(A) &= A(b_{2m+1}A^{2m} + \cdots + b_3A^2 + b_1I) + b_{2m}A^{2m} + \cdots + b_2A^2 + b_0I \\ &=: U + V, \end{aligned} \tag{2.9}$$

quindi p_{2m+1} e $q_{2m+1} = -U + V$ possono essere stimati allo stesso costo di p_{2m} e q_{2m}. Comunque, per $m \geq 12$ questo schema può essere migliorato. Ad esempio, si può scrivere

$$\begin{aligned} p_{12}(A) &= A^6(b_{12}A^6 + b_{10}A^4 + b_8A^2 + b_6I) + b_4A^4 + b_2A^2 + b_0I \\ &\quad + A[A^6(b_{11}A^4 + b_9A^2 + b_7I) + b_5A^4 + b_3A^2 + b_1I] \\ &=: U + V, \end{aligned}$$

e $q_{12}(A) = U - V$. Perciò si possono stimare p_{12} e q_{12} con sole sei moltiplicazioni tra matrici (per A^2, A^4, A^6 e tre moltiplicazioni aggiuntive). Prendendo invece $m = 13$ si

[1] Si intende il prodotto righe per colonne

Tabella 2.2: Numero di moltiplicazioni π_m necessarie per stimare $p_m(A)$ e $q_m(A)$, e misura del costo complessivo C_m in (2.10)

m	1	2	3	4	5	6	7	8	9	10
π_m	0	1	2	3	3	4	4	5	5	6
C_m	25	12	8.1	6.6	5.0	4.9	4.1	4.4	3.9	4.5

m	11	12	13	14	15	16	17	18	19	20	21
π_m	6	6	6	7	7	7	7	8	8	8	8
C_m	4.2	3.8	3.6	4.3	4.1	3.9	3.8	4.6	4.5	4.3	4.2

ottiene una formula analoga, con il prodotto esterno per A trasferito al termine U. Si ottengono formule simili per $m \geq 14$. La Tabella 2.2 riassume il numero di moltiplicazioni di matrici richieste per stimare p_m e q_m, indicato con π_m, per $m = 1 : 21$.

Le informazioni delle Tabelle 2.1 e 2.2 permettono di determinare l'algoritmo ottimale quando $\|A\| \geq \theta_{21}$. Dalla Tabella 2.2 si vede che la scelta è tra $m = 1, 2, 3, 5, 7, 9, 13, 17$ e 21 (in quanto non c'è ragione di usare, ad esempio, $m = 6$ dal momento che il costo per stimare il più preciso q_7 è lo stesso di quello per calcolare q_6). L'aumento da uno di questi valori di m al successivo richiede il prodotto di una matrice in più per calcolare r_m, ma questo viene compensato dal più grande $\theta_m = \|2^{-s}A\|$ consentito se θ_m aumenta più di un fattore 2, poiché diminuendo s a 1 si risparmia una moltiplicazione nella fase finale di quadratura. Di conseguenza la Tabella 2.1 mostra che $m = 13$ è la scelta migliore. Un altro modo per arrivare a questa conclusione è di osservare che il costo dell'algoritmo nelle moltiplicazioni tra matrici è, dato che $s = \lceil \log_2 \|A\|/\theta_m \rceil$ se $\|A\| \geq \theta_m$ e $s = 0$ altrimenti,

$$\pi_m + s = \pi_m + \max(\lceil \log_2 \|A\| - \log_2 \theta_m \rceil, 0),$$

dove $\lceil x \rceil = \min\{n \in \mathbb{Z} : x \leq n\}$ indica la parte intera superiore. Si vuole ora determinare quale m minimizza questa quantità. Per $\|A\| \geq \theta_m$ si può togliere il massimo e ignorare il termine $\|A\|$, che è essenzialmente una costante, e quindi minimizzare

$$C_m = \pi_m - \log_2 \theta_m. \tag{2.10}$$

I valori di C_m sono mostrati nella Tabella 2.2. Di nuovo, si vede che $m = 13$ è ottimale. Si ripetono i calcoli con $u = 2^{-24} \approx 6.0 \times 10^{-8}$ e $u = 2^{-105} \approx 2.5 \times 10^{32}$, che sono

gli epsilon di macchina nello standard IEEE per calcoli, rispettivamente, a precisione singola e a precisione quadrupla; si ha, rispettivamente, che $m = 7$ ($\theta_7 = 3.9$) e $m = 17$ ($\theta_{17} = 3.3$) sono i valori di m ottimali.

Si considerino ora gli effetti degli errori di arrotondamento nel calcolo di $r_m(A)$. Si escludono subito $m = 1$ e $m = 2$, perché r_1 e r_2 possono soffrire della perdita di cifre significative in aritmetica in virgola mobile. Per esempio, r_1 richiede che $\|A\|$ sia di ordine 10^{-8} dopo la scalatura, quindi l'espressione $r_1(A) = (I + A/2)(I - A/2)^{-1}$ perde circa la metà delle cifre significative in A in doppia precisione; ancora, se l'originale matrice A ha norma di ordine almeno 1 allora tutte le cifre significative di qualche elemento di A dovrebbero contribuire al risultato. Si consideri il seguente teorema.

Teorema 2.2. *Il polinomio $\hat{p}_m$ ottenuto applicando il metodo di Horner, l'algoritmo per calcolare un polinomio tramite le potenze esplicite, o il metodo di Paterson-Stockmeyer a p_m, con*

$$p_m(X) = \sum_{k=0}^{m} b_k X^k \text{ con } X \in \mathbb{C}^{n\times n},$$

soddisfa

$$|p_m - \hat{p}_m| \leq \tilde{\gamma}_{mn} \tilde{p}_m(|X|),$$

dove $\tilde{p}_m(|X|) = \sum_{k=0}^{m} |b_k| X^k$. Quindi $\|p_m - \hat{p}_m\|_{1,\infty} \leq \tilde{\gamma}_{mn} \tilde{p}_m(\|X\|_{1,\infty})$.

Applicando il teorema a $p_m(A)$, dove $\|A\|_1 \leq \theta_m$, e notando che i coefficienti di p_m sono tutti positivi, si vede che

$$\begin{aligned}
\|p_m(A) - \hat{p}_m(A)\|_1 &\leq \tilde{\gamma}_{mn} p_m(\|A\|_1) \\
&\approx \tilde{\gamma}_{mn} e^{\|A\|_1/2} \\
&\leq \tilde{\gamma}_{mn} \|e^{A/2}\|_1 e^{\|A\|_1} \\
&\approx \tilde{\gamma}_{mn} \|p_m(A)\|_1 e^{\|A\|_1} \leq \tilde{\gamma}_{mn} \|p_m(A)\|_1 e^{\theta_m}.
\end{aligned}$$

Quindi l'errore relativo è stimato approssimativamente da $\tilde{\gamma}_{mn} e^{\theta_m}$, che è una stima soddisfacente dati i valori di θ_m nella Tabella 2.1. Sostituendo A con $-A$ nell'ultima stima, si ottiene

$$\|q_m(A) - \hat{q}_m(A)\|_1 \leq \tilde{\gamma}_{mn} \|q_m(A)\|_1 e^{\theta_m}.$$

In sintesi, gli errori nel calcolo di p_m e q_m sono ben limitati.

Per ottenere r_m si risolve un sistema lineare con $q_m(A)$ come matrice dei coefficienti, così per avere la certezza di risolvere questo sistema in modo preciso, si deve verificare che $q_m(A)$ sia ben condizionato. È possibile ottenere stime a priori per $\|q_m(A)^{-1}\|$ sotto ipotesi come $\|A\| \leq 1/2$, $\|A\| \leq 1$, o $q_m(-\|A\|) < 2$; tuttavia, queste ipotesi non valgono per ogni m e $\|A\|$ che interessano in questo contesto. Di conseguenza si segue un approccio simile al modo in cui si sono derivate le costanti θ_m.
Con $\|A\| \leq \theta_m$ e scrivendo

$$q_m(A) = e^{-A/2}(I + e^{A/2}q_m(A) - I) =: e^{-A/2}(I + F),$$

si ha, se $\|F\| < 1$,

$$\|q_m(A)^{-1}\| \leq \|e^{A/2}\|\|(I+F)^{-1}\| \leq \frac{e^{\theta_m/2}}{1 - \|F\|}.$$

Si può estendere $e^{x/2}q_m(x) - 1 = \sum_{i=2}^{\infty} d_i x^i$, dalla quale segue $\|F\| \leq \sum_{i=2}^{\infty} |d_i|\theta_m^i$. La stima complessiva è

$$\|q_m(A)^{-1}\| \leq \frac{e^{\theta_m/2}}{1 - \sum_{i=2}^{\infty} |d_i|\theta_m^i}.$$

Determinando simbolicamente i d_i e sommando i primi 150 termini della somma in aritmetica finita a 250 cifre decimali, si ottengono le stime che compaiono nell'ultima riga della Tabella 2.1. Queste stime confermano che q_m è molto ben condizionato per m fino a 13 quando $\|A\| \leq \theta_m$.

A seguire viene esposto l'algoritmo complessivo. Per prima cosa, l'algoritmo controlla se $\|A\| \leq \theta_m$ per $m \in \{3, 5, 7, 9, 13\}$ e, se è così, calcola r_m per i più piccoli m. Altrimenti usa il metodo di scalatura e quadratura con $m = 13$.

Algoritmo 2.3 (Algoritmo di scalatura e quadratura)**.** *Questo algoritmo calcola l'esponenziale di matrice $X = e^A$ di $A \in \mathbb{C}^{n\times n}$ usando il metodo di scalatura e quadratura. Usa le costanti θ_m date nella Tabella 2.1 e i coefficienti di Padé nella Tabella 2.3. L'algoritmo è studiato per lo standard IEEE a doppia precisione.*

1. per $m = [3\ 5\ 7\ 9]$
2. se $\|A\|_1 \leq \theta_m$
% Costruire $r_m(A) =$ *approssimante di Padé di* A *di indici* $[m/m]$.

Tabella 2.3: Coefficienti $b(0:m)$ nel numeratore $p_m(x) = \sum_{i=0}^{m} b_i x^i$ dell'approssimante di Padé $r_m(x)$ di e^x, normalizzati in modo tale che $b(m) = 1$.

m	$b(0:m)$
3	$[120, 60, 12, 1]$
5	$[30240, 15120, 3360, 420, 30, 1]$
7	$[17297280, 8648640, 1995840, 277200, 25200, 1512, 56, 1]$
9	$[17643225600, 8821612800, 2075673600, 302702400,$ $30270240, 2162160, 110880, 3960, 90, 1]$
13	$[64764752532480000, 32382376266240000, 7771770303897600, 1187353796428800,$ $129060195264000, 10559470521600, 670442572800, 33522128640,$ $1323241920, 40840800, 960960, 16380, 182, 1]$

3. *Calcolare U e V usando (2.9) e risolvere $(-U+V)X = U+V$.*

4. *uscire*

5. *fine*

6. *fine*

7. $A \leftarrow A/2^s$ *con* $s \geq 0$ *minimo intero tale che* $\|A/2^s\|_1 \leq \theta_{13}$ *(i.e.,* $s = \lceil \log_2(\|A\|_1/\theta_{13}) \rceil$*).*

8. *% Costruire l'approssimante di Padé di indici* $[13/13]$ *di* e^A*.*

9. $A_2 = A^2$, $A_4 = A_2^2$, $A_6 = A_2 A_4$

10. $U = A[A_6(b_{13}A_6 + b_{11}A_4 + b_9 A_2) + b_7 A_6 + b_5 A_4 + b_3 A_2 + b_1 I]$

11. $V = A_6(b_{12}A_6 + b_{10}A_4 + b_8 A_2) + b_6 A_6 + b_4 A_4 + b_2 A_2 + b_0 I$

12. *Risolvere* $(-U+V)r_{13} = U+V$ *per* r_{13}*.*

13. $X = r_{13}^{2^s}$ *tramite quadrature ripetute.*

Costo: $(\pi_m + \lceil \log_2(\|A\|/\theta_m) \rceil)M + D$, dove m è il grado dell'approssimante di Padé usato, π_m è elencato nella Tabella 2.2, M indica il numero di moltiplicazioni tra matrici e D il numero di "divisioni tra matrici", cioè la soluzione di un sistema lineare del tipo $AX = B$.

Si verifica facilmente che le successioni $\theta_{13}^{2k} b_{2k}$ e $\theta_{13}^{2k+1} b_{2k+1}$ approssimativamente decrescono in modo monotono per k, e di conseguenza l'ordinamento dato nell'Algoritmo 2.3 per il calcolo di U e V prende i termini nell'aumento dell'ordine della norma. Si preferisce questo quando A ha elementi non negativi e non può esserci un cattivo ordinamento, perché non possono esserci tanti annullamenti nelle somme.

La parte dell'algoritmo più sensibile agli errori di arrotondamento è la fase finale di scalatura. Il seguente risultato generale, nel quale si pensa B come approssimante di Padé, mostra il perché.

Teorema 2.4. *Per $B \in \mathbb{R}^{n\times n}$ sia $\hat{X} = fl(B^{2^k})$ calcolato da quadrature ripetute. Allora, per la norma 1, la norma ∞ e la norma di Frobenius,*

$$\|B^{2^k} - \hat{X}\| \leq (2^k - 1)nu\|B\|^2 \cdot \|B^2\|\|B^4\| \cdots \|B^{2^{k-1}}\| + \mathcal{O}(u^2). \tag{2.11}$$

Dimostrazione. La dimostrazione è per induzione. Si ha

$$\|B^2 - fl(B^2)\| \leq \gamma_n\|B\|^2 = nu\|B\|^2 + \mathcal{O}(u^2),$$

quindi la tesi è vera per $k = 1$. Supponendo che il risultato sia vero per $k-1$ e scrivendo $\hat{X}_k = fl(B^{2^k}) =: B^{2^k} + E_k$, si ha

$$\hat{X}_k = fl(\hat{X}_{k-1}^2) = \hat{X}_{k-1}^2 + F_k, \qquad \|F_k\| \leq \gamma_n\|\hat{X}_{k-1}\|^2.$$

Quindi,

$$\begin{aligned}\hat{X}_k &= (B^{2^{k-1}} + E_{k-1})^2 + F_k \\ &= B^{2^k} + B^{2^{k-1}}E_{k-1} + E_{k-1}B^{2^{k-1}} + E_{k-1}^2 + F_k.\end{aligned}$$

Di conseguenza,

$$\begin{aligned}\|E_k\| &\leq 2\|B^{2^{k-1}}\|\|E_{k-1}\| + \|E_{k-1}\|^2 + \|F_k\| \\ &\leq 2(2^{k-1} - 1)nu\|B^{2^{k-1}}\|\|B\|^2\|B^2\|\|B^4\| \cdots \|B^{2^{k-2}}\| + \gamma_n\|\hat{X}_{k-1}\|^2 + \mathcal{O}(u^2) \\ &= 2(2^{k-1} - 1)nu\|B\|^2\|B^2\|\|B^4\| \cdots \|B^{2^{k-1}}\| + \gamma_n\|B^{2^{k-1}}\|^2 + \mathcal{O}(u^2) \\ &\leq (2^k - 1)nu\|B\|^2\|B^2\|\|B^4\| \cdots \|B^{2^{k-1}}\| + \mathcal{O}(u^2).\end{aligned}$$

□

Se si calcola B^{2^k} tramite una serie di moltiplicazioni, la stima dall'alto in (2.11) sarebbe $2^k nu\|B\|^{2^k} + \mathcal{O}(u^2)$. Questa stima però è molto più debole di quella in (2.11), quindi scalare ripetutamente risulta più preciso e anche più efficiente di moltiplicare ripetutamente. Per vedere che (2.11) può comunque essere insoddisfacente, la si riscrive come stima dell'errore relativo

$$\frac{\|B^{2^k} - \hat{X}\|}{\|B^{2^k}\|} \leq \mu(2^k - 1)nu + \mathcal{O}(u^2), \tag{2.12}$$

dove

$$\mu = \frac{\|B\|^2\|B^2\|\|B^4\|\cdots\|B^{2^{k-1}}\|}{\|B^{2^k}\|} \geq 1. \tag{2.13}$$

Il rapporto μ può essere arbitrariamente grande, perché l'annullamento può causare una potenza intermedia B^{2^j} $(j < k)$ molto più grande della potenza finale B^{2^k}. In altre

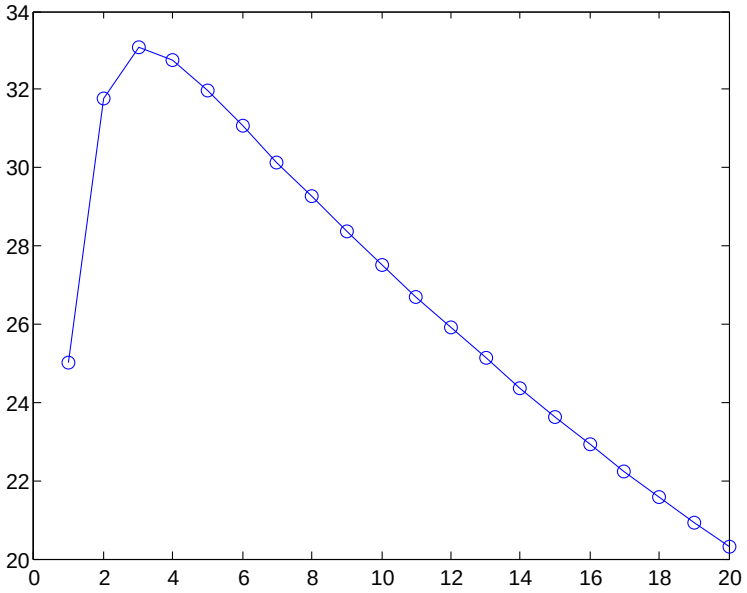

Figura 2.1: Norme 2 delle prime 20 potenze di A in (2.14).

parole, dalle potenze si può vedere il cosiddetto *hump phenomenon* ("hump" in inglese vuol dire gobba, per via del grafico disegnato dalle potenze), illustrato in Figura 2.1 per la matrice

$$A = \begin{pmatrix} -0.97 & 25 \\ 0 & -0.3 \end{pmatrix}. \tag{2.14}$$

Si vede che sebbene le potenze decrescano a zero (dato che $\rho(A) = 0.97 < 1$), inizialmente crescono in norma, creando la "gobba" nel grafico. La gobba può essere arbitrariamente alta in relazione al punto di partenza $\|A\|$. Inoltre, potrebbe esserci più di una sola gobba e infatti per alcune matrici si può osservare ciò che viene chiamato "scalloping behaviour" (letteralmente "comportamento dentellato").

Un altro modo per vedere questa discussione è tramite la curva $\|e^{At}\|$, mostrata in Figura 2.2 per la matrice (2.14). Anche questa curva può avere la forma di una gobba.

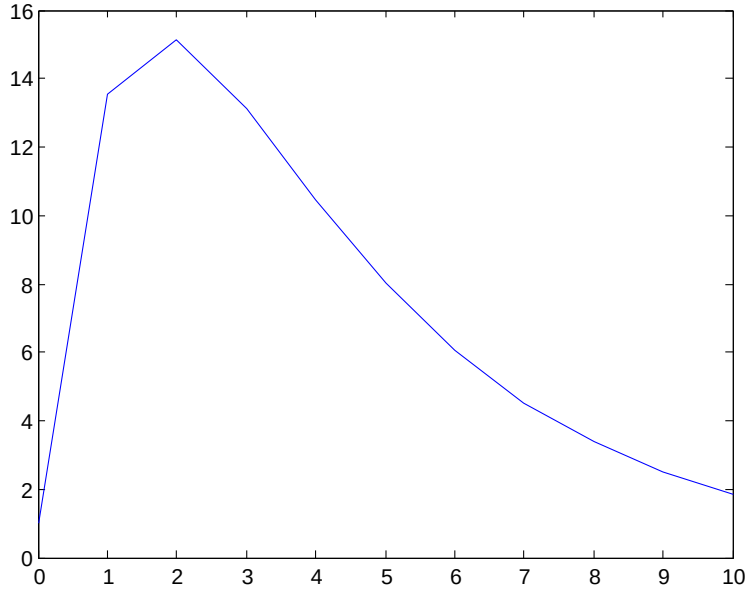

Figura 2.2: Norma 2 di e^{At} per A in (2.14).

Ricordando che si sta usando la relazione $e^A = (e^{A/\sigma})^\sigma$, e che se $t = 1/\sigma$ cala sotto alla gobba ma se $t = 1$ è al di là di essa, allora $\|e^A\| \ll \|e^{A/\sigma}\|^\sigma$. La relazione tra le potenze e le esponenziali è che se si fissa $B = e^{A/r}$ allora i valori $\|B^j\|$ sono i punti sulla curva $\|e^{At}\|$ per $t = 1/r, 2/r, \ldots$.

La stima (2.12) contiene anche il fattore $2^k - 1$. Se $\|A\| > \theta_{13}$ allora $2^k \approx \|A\|$ e di conseguenza la stima dell'errore relativo complessivo contiene un termine della forma $\mu\|A\|nu$. Tuttavia, il fattore $\|A\|$ non è un problema, dal momento che $\|A\|$ è limitato, come conseguenza del Lemma 1.4.

La conclusione è che l'effetto complessivo degli errori di arrotondamento nella fase finale di quadratura può essere grande rispetto all'esponenziale $\tilde{X}$ calcolata, e quindi $\tilde{X}$ può avere un errore relativo grande. Questo può o meno indicare l'instabilità dell'algoritmo, in base al condizionamento del problema e^A per la matrice A. Dal momento che si sa poco circa la dimensione del numero di condizionamento $\kappa_{\texttt{exp}}$ per matrici A non normali, non si possono trarre conclusioni generali chiare sulla stabilità dell'algoritmo.

Nel caso speciale in cui A sia normale, si ha la garanzia che il metodo di scalatura e quadratura è stabile in avanti. Per matrici normali non ci sono gobbe perché $\|A^k\|_2 = \|A\|_2^k$, quindi $\mu = 1$ in (2.13), e $2^k \approx \|A\|_2 = \kappa_{\texttt{exp}}(A)$ per il Teorema 1.5.

La fase di quadratura, quindi, è innocua e l'errore nell'esponenziale calcolata è in accordo con il condizionamento del problema. Un altro caso in cui il metodo di scalatura e quadratura è stabile in avanti è quando $a_{ij} \geq 0$ per $i \neq j$. Il motivo è che l'esponenziale di una matrice del genere è non negativa e, moltiplicando per matrici non negative, si ottiene un metodo stabile, in quanto non ci sono annullamenti.

Si noti, infine, che il metodo di scalatura e quadratura ha una debolezza quando applicato a matrici triangolari a blocchi. Si prenda $A = \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix}$. Com'è fatta la matrice e^A? Si ricordi l'equazione (1.2).

Teorema 2.5 (Teorema di Kenney e Laub)**.** *Sia f una funzione di classe C^{2n-1} e siano*

$$A = \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix}, \quad D = \begin{pmatrix} A_{11} & 0 \\ 0 & A_{22} \end{pmatrix}, \quad N = \begin{pmatrix} 0 & A_{12} \\ 0 & 0 \end{pmatrix}.$$

Allora $f(A) = f(D) + L(D, N)$.

Dimostrazione. Se f è un polinomio di grado m allora il risultato è banale se $m \leq 1$ e altrimenti segue dal fatto che $f(D + N) - f(D) - L(D, N)$ comprende le somme dei prodotti della forma $X_1, X_2, \cdots, X_p$, $p \geq 2$, dove ogni X_i è o D o N e almeno 2 degli X_i sono N. Ognuno di questi prodotti è zero. Per una qualsiasi f, si sa che la derivata di Fréchet $L(D, N)$ di f è la stessa del polinomio $p_{D \oplus D}$ che interpola f. Da ciò segue la tesi. □

Considerando il teorema appena enunciato e la Definizione 1.6, si può costruire e^A. La forma dei blocchi sulla diagonale è immediata. Il blocco $(1, 2)$ invece corrisponde al blocco $(1, 2)$ di

$$\int_0^1 \begin{pmatrix} e^{A_{11}(1-s)} & 0 \\ 0 & e^{A_{22}(1-s)} \end{pmatrix} \begin{pmatrix} 0 & A_{12} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} e^{A_{11}s} & 0 \\ 0 & e^{A_{22}s} \end{pmatrix} ds.$$

Allora

$$e^A = \begin{pmatrix} e^{A_{11}} & \displaystyle\int_0^1 e^{A_{11}(1-s)} A_{12} e^{A_{22}s} ds \\ 0 & e^{A_{22}} \end{pmatrix}.$$

La dipendenza lineare del blocco $(1, 2)$ di e^A da A_{12} suggerisce che la precisione del corrispondente blocco dell'approssimante di Padé non dovrebbe essere gravemente influenzato

da $\|A_{12}\|$ e, quindi, che nel metodo di scalatura e quadratura solo le norme di A_{11} e A_{22} dovrebbero influenzare la quantità di scalatura (indicata da s nell'Algoritmo 2.3). Ma dato che s dipende complessivamente dalla norma di A, quando $\|A_{12}\| \gg \max(\|A_{11}\|, \|A_{22}\|)$ i blocchi diagonali sono scalati in relazione al calcolo di $e^{A_{11}}$ e $e^{A_{22}}$, e ciò può avere un effetto dannoso sulla precisione dell'esponenziale calcolata. Di fatto, il caso del blocco triangolare merita un trattamento speciale. Se gli spettri di A_{11} e A_{22} sono ben separati, allora la cosa migliore è calcolare separatamente $e^{A_{11}}$ e $e^{A_{22}}$ e ottenere F_{12} dalla ricorrenza del blocco di Parlett e risolvendo un'equazione di Sylvester. In generale, un'analisi fatta da Dieci e Papini suggerisce che se il metodo di scalatura e quadratura viene usato con s determinato in modo tale che $2^{-s}\|A_{11}\|$ e $2^{-s}\|A_{22}\|$ sono stimati in modo adeguato, senza considerare $\|A_{12}\|$, allora si otterrà ancora un'approssimazione precisa di e^A.

Capitolo 3

Algoritmi di Schur

Per calcolare e^A si può utilizzare un altro metodo: la decomposizione di Schur $A = QTQ^H$, dove Q è una matrice unitaria e T una matrice triangolare superiore le cui entrate diagonali sono esattamente gli autovalori di A. In questo modo il problema si riduce al calcolo dell'esponenziale di una matrice triangolare. Nel seguito vengono esposti tre principali algoritmi per calcolare e^T.

3.1 Interpolazione alle differenze divise di Newton

Parlett e Ng hanno sviluppato un algoritmo basato sulla decomposizione di Schur e sulla ricorrenza di Parlett che è adattato all'esponenziale in entrambi i suoi blocchi (uno schema a due livelli) e a come si calcola l'esponenziale dei blocchi diagonali. Ci si concentrerà nel seguito su quest'ultimo aspetto.

Definizione 3.1. Siano $x_0, x_1, \ldots, x_n \in \mathbb{C}$ ordinati in modo tale che se $x_i = x_j$ $(i < j)$ allora $x_i = x_{i+1} = \ldots = x_j$. Le *differenze divise* di una funzione f rispetto ai punti x_k si definiscono ricorsivamente in questo modo:

$$f[x_k] = f(x_k),$$

$$f[x_k, x_{k+1}] = \begin{cases} \dfrac{f(x_{k+1}) - f(x_k)}{x_{k+1} - x_k}, & x_k \neq x_{k+1}, \\ f'(x_{k+1}), & x_k = x_{k+1}, \end{cases}$$

$$f[x_0, x_1, \ldots, x_{k+1}] = \begin{cases} \dfrac{f[x_1, x_2, \ldots, x_{k+1}] - f[x_0, x_1, \ldots, x_k]}{x_{k+1} - x_0}, & x_0 \neq x_{k+1}, \\ \dfrac{f^{k+1}(x_{k+1})}{k+1!}, & x_0 = x_{k+1} \end{cases} \tag{3.1}$$

Per calcolare l'esponenziale di un blocco $T_{ii} \in \mathbb{C}^{m\times m}$, Parlett e Ng usano la forma delle differenze divise di Newton per l'interpolazione polinomiale di Hermite

$$p(t) = \sum_{i=1}^{m} c_i \prod_{j=1}^{i-1} (t - \lambda_j), \tag{3.2}$$

dove $c_i = f[\lambda_1, \lambda_2, \ldots, \lambda_i]$ con $\lambda_i \equiv t_{ii}$ e $f(t) = e^t$. Il costo per stimare $p(T_{ii})$ è $\mathcal{O}(m^4)$ flops.

Per funzioni generiche f (o un insieme di valori di funzione dati), le differenze divise sono calcolate usando la ricorrenza standard in (3.1). Tuttavia la ricorrenza può produrre risultati imprecisi in aritmetica in virgola mobile. Questo lo si può vedere dal primo ordine delle differenze divise $f[\lambda_k, \lambda_{k+1}] = (f(\lambda_{k+1}) - f(\lambda_k))/(\lambda_{k+1} - \lambda_k)$ $(\lambda_k \neq \lambda_{k+1})$, in cui per λ_k vicino a λ_{k+1} la sottrazione a numeratore risentirà della cancellazione e l'errore risultante verrà amplificato da un piccolo denominatore. È importante ottenere differenze divise precise poiché i termini del prodotto della matrice in (3.2) possono variare enormemente in norma. Per una particolare f data in forma funzionale piuttosto che come semplici valori di funzione $f(\lambda_i)$, si spera di riuscire ad ottenere le differenze divise in modo più preciso sfruttando le proprietà di f. Il seguente teorema, che mostra come si possono ottenere le differenze divise nella prima riga valutando f con una matrice bidiagonale, offre un modo per ottenere un risultato più preciso.

Teorema 3.1 (Teorema di Opitz)**.** *La differenza divisa $f[\lambda_1, \lambda_2, \ldots, \lambda_m]$ è l'elemento $(1, m)$ di $f(Z)$, dove*

$$Z = \begin{pmatrix} \lambda_1 & 1 & & & \\ & \lambda_2 & 1 & & \\ & & \ddots & \ddots & \\ & & & \ddots & 1 \\ & & & & \lambda_m \end{pmatrix}.$$

Per $m = 2$ il teorema è

$$f\left(\begin{pmatrix} \lambda_1 & 1 \\ 0 & \lambda_2 \end{pmatrix}\right) = \begin{pmatrix} f(\lambda_1) & \frac{f(\lambda_2) - f(\lambda_1)}{\lambda_2 - \lambda_1} \\ 0 & f(\lambda_2) \end{pmatrix},$$

mentre per $\lambda_1 = \lambda_2 = \ldots = \lambda_m$ si ottiene la formula dei blocchi di Jordan

$$f(Z) = \begin{pmatrix} f(\lambda_k) & f'(\lambda_k) & \ldots & \frac{f^{(m_k-1)}(\lambda_k)}{(m_k-1)!} \\ & f(\lambda_k) & \ddots & \vdots \\ & & \ddots & f'(\lambda_k) \\ & & & f(\lambda_k) \end{pmatrix},$$

considerato che $f[\underbrace{x, x, \ldots, x}_{k+1 volte}] = \frac{f^{(k)}(x)}{k!}$.

McCurdy, Ng e Parlett hanno studiato in dettaglio la stima precisa delle differenze divise dell'esponenziale. Sono quindi arrivati ad un algoritmo ibrido che usa la ricorrenza standard quando il denominatore non è troppo piccolo, altrimenti utilizza la serie di Taylor dell'esponenziale insieme al Teorema 3.1 in un sofisticato modo che usa il metodo di scalatura e quadratura. Hanno anche discusso sull'uso della riduzione della matrice, che porta alcuni vantaggi al loro argoritmo in quanto può ridurre la dimensione delle parti immaginarie degli autovalori. Si noti che se $J_k(\lambda)$ è un blocco di Jordan, allora $\texttt{exp}(J_k(\lambda)) = \texttt{exp}(J_k(\lambda - 2\pi ij))$ per ogni j, in quanto nella formula dei blocchi di Jordan i valori di e^t e le derivate non cambiano con la scalatura $\lambda \to \lambda - 2\pi ij$). Quindi per una matrice arbitraria A, ogni autovalore può essere scalato da un multiplo intero di $2\pi i$ così che la sua parte immaginaria sia al massimo di modulo π senza cambiare e^A. Se A è in forma triangolare allora si può usare una tecnica sviluppata da Ng e basata sulla ricorrenza di Parlett per eseguire la riduzione della matrice. Li ha mostrato come sia possibile stimare (3.2), quando A è reale, principalmente in aritmetica reale, con aritmetica complessa limitata al calcolo delle differenze divise.

Purtroppo nei riferimenti sopra citati non vi sono precise asserzioni di un algoritmo complessivo per e^A e non contengono test numerici accurati.

3.2 Algoritmo di Schur-Fréchet

Kenney e Laub hanno sviluppato un algoritmo basato sul Teorema 2.5.

Questo algoritmo mostra che se $T = \begin{pmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{pmatrix}$ allora il blocco $(1,2)$ di e^T è il blocco $(1,2)$ della derivata di Fréchet $L\left(diag(T_{11}, T_{22}), \begin{pmatrix} 0 & T_{12} \\ 0 & 0 \end{pmatrix}\right)$.
Si calcola la derivata di Fréchet usando l'algoritmo descritto nel seguito.

Teorema 3.2 (Teorema di rappresentazione di Kronecker della derivata di Fréchet)**.** *Per $A \in \mathbb{C}^{n\times n}$, $vec(L(A,E)) = K(A)vec(E)$, dove $K(A) \in \mathbb{C}^{n^2\times n^2}$ ha le seguenti rappresentazioni*

$$K(A) = \begin{cases} (I \otimes e^A)\psi_1(A^T \oplus (-A)), \\ (e^{A^T/2} \otimes e^{A/2})\,\mathrm{sinch}(\frac{1}{2}[A^T \oplus (-A)]), \\ \frac{1}{2}(e^{A^T} \oplus e^A)\tau\left(\frac{1}{2}[A^T \oplus (-A)]\right), \end{cases} \tag{3.3}$$

dove $\psi_1(x) = (e^x - 1)/x$ e $\tau(x) = \tanh(x)/x$. La terza espressione è valida se $\frac{1}{2}\|[A^T \oplus (-A)]\| < \pi/2$ per ogni norma compatibile.

Un modo per approssimare $L(A,E)$ è usando la formula di Kronecker (3.3). Si consideri la formula

$$vec(L(A,E)) = \frac{1}{2}(e^{A^T} \oplus e^A)\tau\left(\frac{1}{2}[A^T \oplus (-A)]\right) vec(E), \tag{3.4}$$

dove $\tau(x) = \tanh(x)/x$ e si assume che $\|\frac{1}{2}[A^T \oplus (-A)]\| < \pi/2$. Nella formula si ha che per una matrice $A \in \mathbb{C}^{m\times n}$, $A = [a_1, a_2, \ldots, a_m]$, $vec(A) = [a_1^T, a_2^T, \ldots, a_m^T]^T$ e vale $vec(AXB) = (B^T \otimes A)vec(X)$.

Allora si ha che

$$L(A,E) = \frac{1}{2}(Ye^A + e^A Y), \qquad vec(Y) = \tau\left(\frac{1}{2}[A^T \oplus (-A)]\right) vec(E).$$

Si noti che $\frac{1}{2}[A^T \oplus (-A)]vec(E) = \frac{1}{2}(A^T \otimes I - I \otimes A)vec(E) = \frac{1}{2}vec(EA - AE)$. Quindi se $r(x)$ è un approssimazione razionale di $\tau(x)$ in cui sia numeratore che denominatore sono fattorizzati in fattori lineari allora $r\left(\frac{1}{2}[A^T \oplus (-A)]\right) vec(E)$ può essere stimato al costo della risoluzione di una sequenza di equazioni di Sylvester contenenti "prodotti di Sylvester" sul lato destro. Per essere precisi, sia

$$\tau(x) \approx r_m(x) = \prod_{i=1}^{m} (x/\beta_i - 1)^{-1}(x/\alpha_i - 1)$$

un'approssimazione razionale. Allora si può approssimare $L(A,E)$ usando la seguente "cascata di Sylvester":

$$G_0 = E,$$
$$\left(I + \frac{A}{\beta_i}\right) G_i + G_i \left(I - \frac{A}{\beta_i}\right) = \left(I + \frac{A}{\alpha_i}\right) G_{i-1} + G_{i-1} \left(I - \frac{A}{\alpha_i}\right), \quad i = 1:m,$$
$$L(A,E) \approx \frac{1}{2}(G_m e^A + e^A G_m).$$

È naturale scegliere come r_m un approssimante di Padé. Gli approssimanti di Padé di τ si possono ottenere troncando lo sviluppo continuo delle frazioni

$$\tau(x) = \tanh(x)/x = 1 + \cfrac{1}{1 + \cfrac{x^2/(1\cdot 3)}{1 + \cfrac{x^2/(3\cdot 5)}{1 + \ldots + \cfrac{x^2/((2k-1)\cdot(2k+1))}{1 + \cdots}}}}.$$

Kenney e Laub hanno mostrato che per $\|A\| < \pi/2$ si ha $\|\tau(A) - r_m(A)\| \le g(\|A\|)$, dove $g(x) = \tau(ix) - r_m(ix)$ e la norma è una qualsiasi norma compatibile. Definendo $C = \frac{1}{2}[A^T \oplus (-A)]$ e notando che $\|C\|_p \le \|A\|_p$ per $1 \le p \le \infty$, segue che se $\|A\|_p \le 1$ allora $\|\tau(C) - r_8(C)\|_p \le g(1) = 1.44 \times 10^{-16}$, e viene soddisfatta la condizione $\|C\|_p \le \pi/2$ affinché (3.4) sia valida. La condizione $\|A\|_p \le 1$ può essere sistemata scalando $A \leftarrow 2^{-s}A$ e usando la ricorrenza

$$L_s = L_{\texttt{exp}}(2^{-s}A, 2^{-s}E),$$
$$L_{i-1} = e^{2^{-i}A} L_i + L_i e^{2^{-i}A}, \qquad i = s:-1:1$$

per annullare gli effetti della scalatura. L'approssimante di Padé di indici [8/8] è dato da

$$r_8(x) = \frac{p_8(x)}{q_8(x)} = \frac{x^8 + 990x^6 + 135135x^4 + 4729725x^2 + 34459425}{45x^8 + 13860x^6 + 945945x^4 + 16216200x^2 + 34459425}$$

e gli zeri di p_8 e di q_8 sono dati nella Tabella 3.1. Nella scelta della scalatura di A si deve considerare anche il condizionamento della cascata di Sylvester. Questo si riduce ad assicurarsi che il limite superiore in

$$\prod_{i=1}^{m} \|(C/\beta_i - I)^{-1}\|_p \|C/\alpha_i - I\|_p \le \prod_{i=1}^{m} \frac{1 + \|C/\alpha_i\|_p}{1 - \|C/\beta_i\|_p} \le \prod_{i=1}^{m} \frac{1 + \|A/\alpha_i\|_p}{1 - \|A/\beta_i\|_p}$$

Tabella 3.1: Zeri α_j del numeratore p_s e β_j del denominatore q_s dell'approssimante di Padé di indici [8/8] r_s di $\tau = \tanh(x)/x$, con 5 cifre significative.

α_j	β_j
$\pm 3.1416e0i$	$\pm 1.5708e0i$
$\pm 6.2900e0i$	$\pm 4.7125e0i$
$\pm 1.0281e1i$	$\pm 7.9752e0i$
$\pm 2.8894e1i$	$\pm 1.4823e1i$

non sia troppo grande. Per $m = 8$ e la norma 1, questa stima è al massimo 55.2 se $\|A\|_p \leq 1$, che è abbastanza accettabile.

Il seguente algoritmo raccoglie le idee espresse sopra.

Algoritmo 3.3 (Algoritmo della derivata di Fréchet dell'esponenziale di matrice)**.** *Data $A \in \mathbb{C}^{n\times n}$, l'algoritmo calcola la derivata di Fréchet $L = L(A, E)$ dell'esponenziale di matrice tramite* (3.4)*, usando il metodo di scalatura e quadratura e l'approssimante di Padé di indici* [8/8] *a $\tau = \tanh(x)/x$. Sfrutta le costanti nella Tabella 3.1. L'algoritmo è studiato per lo standard IEEE a doppia precisione.*

1. $B = A/2^s$ *con* $s \geq 0$ *il minimo intero tale che* $\|A/2^2\|_1 \leq 1$.
2. $G_0 = 2^{-s}E$
3. per $i = 1 : 8$
4. *Risolvere per* G_i *l'equazione di Sylvester* $(I + B/\beta_i)G_i + G_i(I - B/\beta_i) = (I + B/\alpha_i)G_{i-1} + G_{i-1}(I - B/\alpha_i)$.
5. fine
6. $X = e^B$
7. $L_s = (G_8X + XG_8)/2$
8. per $i = s : -1 : 1$
9. *se* $i < s$, $X = e^{2^{-i}A}$, *fine*
10. $L_{i-1} = XL_i + L_iX$
11. fine
12. $L = L_0$

Costo: $(18+2s)M$ e s esponenziali di matrici (o 1 esponenziale e $(17+3s)M$ se si usa la quadratura ripetuta alla riga 9), e la soluzione di 8 equazioni di Sylvester.

In pratica si usa una decomposizione di Schur iniziale di A per ridurre il costo. Possono essere sviluppati algoritmi simili per le altre due formule in (3.3).

Per un algoritmo generale, $e^{T_{11}}$ e $e^{T_{22}}$ vengono ricorsivamente ridotte in blocchi e calcolate allo stesso modo.

3.3 Algoritmo di Schur-Parlett

Per ottenere un algoritmo specializzato per l'esponenziale di matrice basta una semplice modifica all'algoritmo di Schur Parlett, richiamando l'Algoritmo 2.3 di scalatura e quadratura per i blocchi diagonali visto nel capitolo precedente. È importante quindi calcolare l'esponenziale di una qualsiasi matrice 2×2 diagonale a blocchi $\begin{pmatrix}\lambda_1 & t_{12}\\ 0 & \lambda_2\end{pmatrix}$ da una formula esplicita.

Per una matrice triangolare superiore 2×2 e per una funzione f definita sullo spettro della matrice, si ha che

$$f\left(\begin{pmatrix}\lambda_1 & t_{12}\\ 0 & \lambda_2\end{pmatrix}\right)=\begin{pmatrix}f(\lambda_1) & t_{12}\frac{f(\lambda_2)-f(\lambda_1)}{\lambda_2-\lambda_1}\\ 0 & f(\lambda_2)\end{pmatrix}. \tag{3.5}$$

Si può vedere che l'elemento $(1,2)$ della formula subisce la cancellazione quando $\lambda_1\approx\lambda_2$. Tuttavia, considerando la funzione esponenziale, si può riscrivere la formula come

$$\begin{aligned} t_{12}\frac{e^{\lambda_2}-e^{\lambda_1}}{\lambda_2-\lambda_1} &= t_{12}e^{(\lambda_1+\lambda_2)/2}\frac{e^{(\lambda_2-\lambda_1)/2}-e^{(\lambda_1-\lambda_2)/2}}{\lambda_2-\lambda_1}\\ &= t_{12}e^{(\lambda_1+\lambda_2)/2}\frac{\sinh((\lambda_2-\lambda_1)/2)}{(\lambda_2-\lambda_1)/2}. \end{aligned}$$

Di conseguenza si può riscrivere la formula, valida per ogni λ_1 e λ_2, come

$$\exp\left(\begin{pmatrix}\lambda_1 & t_{12}\\ 0 & \lambda_2\end{pmatrix}\right)=\begin{pmatrix}e^{\lambda_1} & t_{12}e^{(\lambda_1+\lambda_2)/2}\operatorname{sinch}((\lambda_1-\lambda_2)/2)\\ 0 & e^{\lambda_2}\end{pmatrix}, \tag{3.6}$$

dove $\operatorname{sinch}((\lambda_1-\lambda_2)/2)=\dfrac{\sinh((\lambda_2-\lambda_1)/2)}{(\lambda_2-\lambda_1)/2}$. Questa fornirà un risultato preciso fintanto che sarà disponibile un'implementazione precisa di sinch.

Algoritmo 3.4 (Algoritmo dello schema dei blocchi)**.** *Data una matrice triangolare* $T \in \mathbb{C}^{n\times n}$ *con autovalori* $\lambda_i \equiv t_{ii}$ *e un parametro di blocco* $\delta > 0$*, questo algoritmo produce uno schema a blocchi, definito da un vettore intero* q*, per la ricorrenza dei blocchi di Parlett: l'autovalore* λ_i *viene assegnato ad un insieme* S_{q_i}*, e soddisfa le condizioni che* $\min\{|\lambda_i - \lambda_j| : \lambda_i \in S_p, \lambda_j \in S_q, p \neq q\} > \delta$*, e per ogni insieme* S_i *con più di un elemento, ogni elemento di* S_i *è entro la distanza massima* δ *da qualche altro elemento dell'insieme. Per ogni insieme* S_q*, tutti gli autovalori in* S_q *compariranno insieme nel blocco triangolare superiore* $\tilde{T}_{ii}$ *di* $\tilde{T} = U^*TU$*.*

1. $p = 1$
2. Inizializzare S_p *come insieme vuoto.*
3. per $i = 1 : n$
4. *se* $\lambda_i \notin S_q$ *per ogni* $1 \leq q < p$
5. *Assegnare* λ_i *a* S_p*.*
6. $p = p + 1$
7. *fine*
8. *per* $j = i + 1 : n$
9. *Denotare con* S_{q_i} *l'insieme che contiene* λ_i*.*
10. *se* $\lambda_j \notin S_{q_i}$
11. *se* $|\lambda_i - \lambda_j| \leq \delta$
12. *se* $\lambda_j \notin S_k$ *per ogni* $1 \leq k < p$
13. *Assegnare* λ_j *a* S_{q_i}*.*
14. *altrimenti*
15. *Spostare gli elementi di* $S_{\max(q_i,q_j)}$ *a* $S_{\min(q_i,q_j)}$*.*
16. *Ridurre di 1 gli indici degli insiemi* S_q *per* $q > \max(q_i, q_j)$*.*
17. $p = p - 1$
18. *fine*
19. *fine*
20. *fine*
21. *fine*
22. fine

Algoritmo 3.5 (Algoritmo di Schur-Parlett per l'esponenziale di matrice)**.** *Data* $A \in \mathbb{C}^{n\times n}$*, questo algoritmo calcola* $F = e^A$ *tramite la decomposizione di Schur.*

1. Calcolare la decomposizione di Schur $A = QTQ^*$ *(*Q *unitaria,* T *triangolare superiore)*
2. Se T *è diagonale,* $F = e^T$*, andare alla riga 12, fine*
3. Usando l'Algoritmo 3.4 con parametro di blocco $\delta = 0.1$*, assegnare ogni autovalore* λ_i *ad un insieme* S_{q_i}*.*
4. Scegliere una permutazione confluente q' *di* q *ordinata per indice medio.*
5. Riordinare T *in base a* q' *e aggiornare* Q*.*
% Ora $A = QTQ^*$ *è la riordinata decomposizione di Schur, con blocco* $m \times m$ T
6. per $i = 1 : m$
7. Calcolare $F_{ii} = e^{T_{ii}}$ *direttamente se* T_{ii} *è* 1×1*, tramite* (3.6) *se* T_{ii} *è* 2×2*, o altrimenti tramite l'Algoritmo di scalatura e quadratura 2.3.*
8. per $j = i - 1 : -1 : 1$
9. Risolvere l'equazione di Sylvester $T_{ii}F_{ij} - F_{ij}T_{jj} = F_{ii}T_{ij} - T_{ij}F_{jj} + \sum_{k=i+1}^{j-1}(F_{ik}T_{kj} - T_{ik}F_{kj})$ *per* F_{ij}*.*
10. fine
11. fine
12. $F = QFQ^*$

Costo: Il costo dell'algoritmo dipende enormemente dalla distribuzione degli autovalori di A, ed è approssimativamente tra i $28n^3$ flops e $n^4/3$ flops. Si noti che Q, e quindi F, può essere tenuta in forma fattorizzata con un significativo risparmio computazionale. Questo è adatto se, ad esempio, si deve applicare F a pochi vettori.

È stato fissato il parametro di blocco $\delta = 0.1$, che gli esperimenti hanno indicato essere una buona scelta di default. La scelta ottimale di δ in termini di costo o precisione dipende dal problema.

L'Algoritmo 3.5 ha una proprietà interessante: agisce semplicemente sui casi semplici. In particolare, se A è normale, così che la decomposizione di Schur sia $A = QDQ^*$ con D diagonale, l'algoritmo calcola semplicemente $e^A = Qe^DQ^*$. D'altra parte, se A ha un solo autovalore di molteplicità n, allora l'algoritmo lavora con un singolo blocco, $T_{11} \equiv T$, e

calcola $e^{T_{11}}$ attraverso il suo sviluppo in serie di Taylor nell'autovalore.

Per matrici reali, potrebbe sembrare che utilizzando la decomposizione di Schur reale nel primo punto dell'Algoritmo 3.5 sia possibile lavorare interamente in aritmetica reale. Tuttavia, la strategia dell'algoritmo di collocare autovalori non vicini in blocchi diversi richiede la divisione di coppie di autovalori coniugati complessi con parti immaginarie grandi, forzando l'aritmetica complessa, quindi l'algoritmo non si presta in genere all'utilizzo della forma di Schur reale. Tuttavia, se A è reale e normale allora la decomposizione di Schur reale è diagonale a blocchi, non è necessaria alcuna riorganizzazione, e l'algoritmo 3.5 può essere ridotto al calcolo della forma di Schur e al calcolo dell'esponenziale sui blocchi diagonali.

Dagli esperimenti numerici sono emersi alcuni aspetti negativi. L'algoritmo può essere instabile, nel senso che l'errore relativo in norma può superare enormemente $\kappa_{\mathtt{exp}}(A)u$. Cambiare il parametro di blocco δ (da 0.1 a 0.2) può produrre un blocco diverso che elimina l'instabilità. Tuttavia, l'instabilità si può presentare per tutte le scelte di δ. Inoltre l'instabilità si può presentare per tutte le partizioni in blocchi non banali (cioè qualsiasi partizione con più di un blocco), alcune delle quali potrebbero non essere create per un'appropriata scelta di δ nell'algoritmo. Quest'ultimo punto indica una debolezza fondamentale della ricorrenza di Parlett.

Il calcolo dell'esponenziale dei blocchi diagonali dell'Algoritmo 2.3 porta la maggior efficienza dell'approssimazione di Padé rispetto all'approssimazione di Taylor per l'esponenziale e sfrutta anche il metodo di scalatura e quadratura (anche se quest'ultima potrebbe essere usato in combinazione con una serie di Taylor).

Rispetto all'Algoritmo 2.3 applicato all'intera matrice (triangolare), la probabilità della sovra scalatura è ridotta perché l'algoritmo viene applicato solo ai blocchi (solitamente di piccole dimensioni) diagonali (e non ai blocchi diagonali 1×1 o 2×2). L'Algoritmo 3.5 è una valida alternativa all'Algoritmo 2.3. Ha il vantaggio rispetto agli altri algoritmi visti della semplicità e maggiore efficienza, e le sue potenziali instabilità sono più chiare.

3.4 Esperimento numerico

Viene qui descritto un esperimento che confronta due metodi per il calcolo di e^A, dove e^A viene calcolata tramite la funzione di MATLAB `expm` che è la funzione più precisa ed affidabile per calcolare numericamente l'esponenziale di matrice. In particolare, `expm` implementa l'Algoritmo 2.3.

Nell'esperimento vengono prese 14 matrici test 10×10 dalla funzione `gallery` di MATLAB con numeri di condizionamento diversi. La Figura 3.1 mostra i numeri di

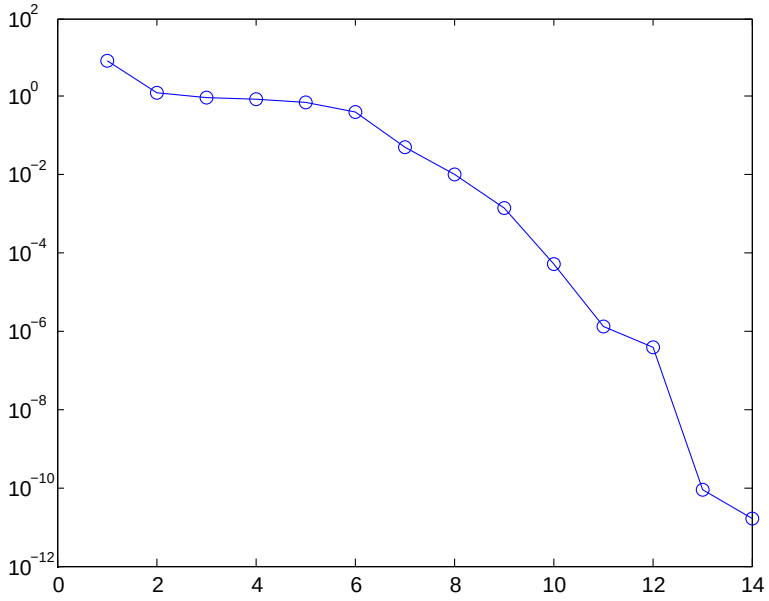

Figura 3.1: Numeri di condizionamento relativo delle matrici test.

condizionamento delle matrici test prese in esame, dove le matrici sono state ordinate per numero di condizionamento decrescente. In particolare, il grafico dei numeri di condizionamento tende a zero, mostrando che la norma di e^A in (1.3) cresce più velocemente rispetto al numeratore.

I metodi che si vogliono confrontare sono:

1. La funzione `funm` di MATLAB che implementa l'Algoritmo 3.5 quando viene chiamata come `funm(A,@exp)`.

2. La funzione `expmdemo1` di MATLAB: una funzione che implementa il metodo di scalatura e quadratura con $m = 6$ e $\|2^{-s}A\|_\infty \leq 0.5$ come criterio di scala. È una versione M-file della funzione `expm` che è stata usata in MATLAB 7 (R14SP3) e versioni precedenti.

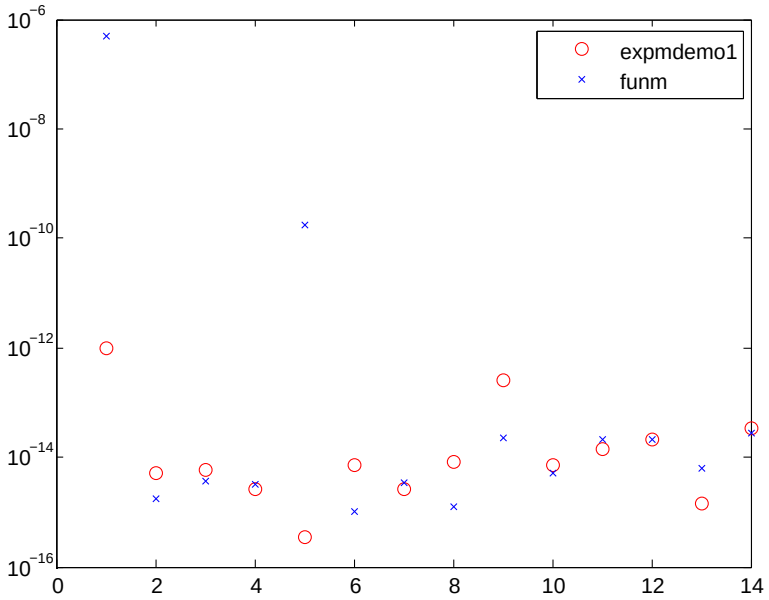

Figura 3.2: Norme di Frobenius degli errori relativi delle funzioni `funm` e `expmdemo1` di MATLAB rispetto alla funzione `expm` per il calcolo dell'esponenziale di matrice.

Costruita in modo analogo alla Figura 3.1, la Figura 3.2 mostra la norma degli errori relativi $\|\hat{X} - \texttt{expm}(A)\|_F / \|\texttt{expm}(A)\|_F$ della $\hat{X}$ calcolata per ogni matrice test A. Gli errori mostrati sono per lo più soddisfacenti, tuttavia si possono notare un paio di eccezioni. La matrice `gallery('chebspec',10)`, prima ad essere rappresentata nelle Figure 3.1 e 3.2, è una matrice simile ad un blocco di Jordan di dimensione 10 con autovalore 0. I suoi autovalori calcolati giacciono approssimativamente in un cerchio di centro 0 e raggio 0.2; per l'Algoritmo 3.5, è la distribuzione più difficile da gestire. Di conseguenza, si ha un errore relativo maggiore rispetto alla funzione `funm`, che implementa l'algoritmo. Si può notare un'altra evidente eccezione in corrispondenza della quinta matrice rappresentata nel grafico. In esame vi è la matrice `gallery('forsythe',10)`, che è un blocco di

Jordan di dimensione 10 con autovalore 0 ad eccezione di un'entrata (10,1) di $u^{\frac{1}{2}}$. La distribuzione degli autovalori calcolati è simile a quella della matrice precedente.

Si può fare un'ultima osservazione su `expmdemo1`. Questa funzione rappresenta un caso particolare del metodo di quadratura e squadratura implementato dalla funzione `expm` rispetto alla quale è stato fatto l'esperimento, di conseguenza sono buoni gli errori relativi ottenuti. In generale però, risulta un metodo meno affidabile rispetto a `funm` e a `expm`. Ciò è dovuto principalmente alla scelta subottimale di m e s; `expm` richiede di solito una s maggiore e quindi richiede un numero minore di scalature.

Bibliografia

[1] [Higham, 2008] Nick Higham, *Function of matrices: theory and computation* SIAM, Filadelfia, 2008.

Ringraziamenti

Ringrazio innanzitutto la mia relatrice, la prof.ssa Valeria Simoncini, per avermi assistito durante la stesura della tesi con grande disponibilità e competenza.

Non sarei riuscito a superare questi anni nello stesso modo senza i miei colleghi, in particolare Alice, Davide, Filippo e Valentina che mi hanno fatto compagnia tra lezioni, imitazioni, gossip e soprattutto partite a briscola.

Un ringraziamento speciale va a Martina che si è rivelata un'amica splendida, che ha ascoltato i miei sfoghi d'ansia ed è stata un'incredibile compagna di viaggio!

Ringrazio anche il gruppo di D&D che mi ha fatto scoprire un gioco bellissimo in cui sono completamente negato.

Non posso non ringraziare Chiara ed Elisa F. che conosco da tutta la vita e, nonostante io le faccia impazzire, continuano a supportarmi (ma soprattutto a sopportarmi).

Grazie a Yuuki che è diventato in poco uno dei migliori amici che potessi trovare, che mi ascolta con incredibile e sconfinata pazienza e nonostante tutto continua a supportarmi e a volermi bene.

Grazie ad Elisa S. che, per citarla, «se ero normale, non eri mia amica».

Ringrazio di cuore i miei nonni, i miei zii e mia cugina per essersi sempre tenuti informati e per avermi sostenuto ogni volta.

Infine voglio ringraziare la mia famiglia. Grazie a mia sorella e a Simone che, nonostante i mille pensieri e problemi per la testa, riescono sempre a trovare un momento per dedicarmi un pensiero. Grazie ai miei genitori che mi hanno sempre lasciato prendere le mie decisioni in autonomia e mi hanno supportato qualsiasi esse fossero. Se sono riuscito a portare a termine questo percorso è soprattutto grazie a voi.