Ripasso di Calcolo Scientifico:

Giulio Del Corso

Queste dispense sono tratte dalle lezioni del Prof. Gemignani e del Prof. Bini del corso di Calcolo Scientifico (2014/2015) dell'università di Pisa.

Non contengono le dimostrazioni dei teoremi svolti ma gli algoritmi e i risultati ottenuti.

In quanto semplici rielaborazioni potrebbero contenere errori, in tal caso se voleste segnalarmeli all'indirizzo giulio.pisa@virgilio.it ve ne sarei grato.

Ringrazio infine per l'aiuto nel compilarli Giulia Bernardini

Indice:

Capitolo 1: Studio degli errori

Idea:
Studiare gli errori degli algoritmi e di approssimazione per determinare quando è preferibile la scelta di un metodo rispetto ad un altro.

Rappresentazione in aritmetica Floating Point:
Ogni numero reale lo rappresentiamo mediante la scrittura:

$$x = \pm\beta^p \sum_{i=1}^{t} d_i\beta^{-i}$$

Dove d_i sono le *cifre*, p è **l'esponente**, $\sum_{i=1}^{t} d_i\beta^{-i}$ è la **mantissa**.

Osservazione:
t è il numero di cifre e dipende dalla precisione da noi scelta, allo stesso modo l'esponente p varia fra dei valori fissati $-m$; M

Problemi:
Se cerchiamo di rappresentare un valore con esponente maggiore di M si parla di **Overflow**.
Se cerchiamo di rappresentare un valore con esponente minore di $-m$ si parla di **Underflow**.
L'errore relativo $\left|\frac{\bar{x}-x}{x}\right|$ è maggiorato da $\beta^{1-t} \coloneqq u$ detta **Precisione di macchina**.

Tipi di errore:

Errore di rappresentazione (Errore relativo):
Errore dato dalla rappresentazione di un numero reale in aritmetica Floating Point.

$\varepsilon_x \coloneqq \left|\frac{\bar{x}-x}{x}\right|$

Equivalente:
$\bar{x} = x(1+\varepsilon_x)$

Errore inerente:
È l'errore intrinseco al problema, se stiamo ricercando il valore di una funzione $f(x)$ dobbiamo limitarci a calcolare il valore di $f(\bar{x})$ con $\bar{x}$ la scrittura di x in Floating Point.

$\varepsilon_{in} \coloneqq \frac{f(\bar{x})-f(x)}{f(x)}$

Osservazione:
Sfruttando lo sviluppo di Taylor possiamo scrivere $\varepsilon_{in} \doteq \varepsilon_x \cdot \frac{xf'(x)}{f(x)}$
$\frac{xf'(x)}{f(x)}$ è detto **Coefficiente di amplificazione**, ε_x è l'**errore di approssimazione**.

Osservazione (Funzioni a più variabili):
$\varepsilon_{in} \doteq \sum_{i=1}^{n} \varepsilon_{x_i} \cdot \frac{x_i \frac{\partial f}{\partial x_i}(x_{1,\dots,n})}{f(x_{1,\dots,x_n})}$

Osservazione:
Prodotto e divisione hanno coefficiente di amplificazione 1.

Errore algoritmico:
È l'errore dato dall'approssimare la funzione f che stiamo calcolando con un numero finito di passaggi con la sua versione approssimata (Sarà funzione degli errori di ogni passaggio).

$\varepsilon_{alg} \coloneqq \frac{\varphi(\bar{x})-f(\bar{x})}{f(\bar{x})}$

Come si calcola:
Si scrivono i vari passaggi dell'algoritmo e si calcola di ciascun passaggio la propagazione dell'errore.
Si segna dunque il coefficiente di amplificazione dato dalla formula precedente e si moltiplica per l'errore del passo precedente.
Si somma infine un errore di rappresentazione.

In conclusione si studia l'errore algoritmico totale provando a maggiorarlo con la precisione di macchina u. L'errore inerente infatti non dipende dall'algoritmo e dunque quello con maggiorazione minore sarà migliore.

Tabella utile:

Operazione:	Coefficiente di amplificazione rispetto al primo termine:
$a+b$	$\frac{a}{a+b}$
$a-b$	$\frac{a}{a-b}$
ab	1
$\frac{a}{b}$	1

Problemi ben condizionati e algoritmi stabili:

Un algoritmo è numericamente stabile se l'errore algoritmico generato è piccolo.

Un problema è ben condizionato se non risente di piccole perturbazioni, ossia se il coefficiente di amplificazione è piccolo.

Errore analitico:

È l'errore dato dall'approssimare una funzione non razionale f con una funzione razionale h calcolabile in un numero finito di passi.

$\varepsilon_{an} \coloneqq \frac{h(x)-f(x)}{f(x)}$

Errore totale:

$\varepsilon_{tot} \coloneqq \frac{\varphi(\bar{x})-f(x)}{f(x)}$

Osservazione pratica:

$\varepsilon_{tot} = \varepsilon_{in} + \varepsilon_{alg} + \varepsilon_{in}\varepsilon_{alg} \doteq \varepsilon_{in} + \varepsilon_{alg}$

Se la funzione da calcolare non è razionale:

$\varepsilon_{tot} \doteq \varepsilon_{in} + \varepsilon_{alg} + \varepsilon_{an}$

Capitolo 2: Algebra lineare numerica

Idea:
Avere dei metodi per determinare in maniera rapida se una matrice è invertibile. Studiare in maniera approssimata gli autovalori così da ottenere maggiorazioni per alcuni numeri di condizionamento.

Definizione (Cerchio di Gershgorin):
Data $A \in M_n(\mathbb{C})$ l'i-esimo cerchio di Gershgorin è definito come:

$$K_i := \{z \in \mathbb{C} \mid |z - a_{i,i}| \leq \sum_{j \neq i} |a_{i,j}|\}$$

Primo teorema di Gershgorin:
Gli autovalori di $A \in M_n(\mathbb{C})$ sono contenuti in $\bigcup_{i=1}^{n} K_i$

Secondo teorema di Gershgorin:
Sia $A \in M_n(\mathbb{C})$ una matrice tale che $\bigcup K_i = M_1 \cup M_2$ unione di cerchi di Gershgorin con $M_1 \cap M_2 = \emptyset$.
Allora M_i contiene tanti autovalori quanti sono i cerchi che lo costituiscono.

Terzo teorema di Gershgorin:
Sia $A \in M_n(\mathbb{C})$ una matrice irriducibile e sia λ un suo autovalore per cui vale la proprietà:

$$\lambda \in K_i \to \lambda \in \delta K_i$$

Allora λ appartiene a tutti i K_i (E quindi alle loro frontiere).

Definizione (Permutazione):
Una matrice P è di permutazione se $p_{i,j} = \delta_{\sigma(i),j}$ e σ permutazione.
Si dice di permutazione perché data A matrice qualsiasi PA è la matrice con le righe permutate.

Vale $PP^T = P^T P = \mathrm{Id}$

Definizione (Riducibile):
Una matrice $A \in M_n(\mathbb{C})$ si dice riducibile se esiste una matrice di permutazione $P \in M_n(\mathbb{C})$ tale che:

$$PAP^T = \begin{pmatrix} A_{1,1} & A_{1,2} \\ 0 & A_{2,1} \end{pmatrix}$$

Definizione (Grafo fortemente connesso):
Un grafo orientato si dice fortemente connesso se per ogni coppia di indici i, j con $i \neq j$ esiste un cammino orientato che parte dal nodo p_i e arriva al nodo p_j.

Teorema:
Una matrice è riducibile se e solo se il grafo orientato associato non è fortemente connesso.

Forma canonica di Schur:

Sia $A \in M_n(\mathbb{C})$, allora esiste una matrice unitaria Q e una matrice triangolare superiore T tale che:

$$A = QTA^H$$

Definizione (Matrice normale):

$A \in M_n(\mathbb{C})$ si dice normale se $AA^H = A^HA$

Teorema:

Una matrice A è normale se e solo se esistono Q unitaria e D diagonale tali che:

$$A = QDQ^H$$

Capitolo 3: Norme vettoriali e matriciali

Idea:
Definizione delle norme vettoriali e matriciali. Definizione di raggio spettrale e studio del condizionamento per stima mediante norma.

Definizione (Norma vettoriale):
Una norma vettoriale è un'applicazione $\| \|: \mathbb{C}^n \to \mathbb{R}$ tale che:

$\|x\| \geq 0\ ;\ \|x\| = 0 \leftrightarrow x = 0$
$\|ax\| = |a| \cdot \|x\|$
$\|x + y\| \leq \|x\| + \|y\|$

Esempi:
$\|x\|_1 = \sum_{i=1}^{n} |x_i|$
$\|x\|_2 = \sqrt{x^H x} = \sqrt{\sum_{i=1}^{n} |x_i|^2}$ (**Norma euclidea**)
$\|x\|_\infty = \max |x_i|$
$\|x\|_p = (\sum |x_i|^p)^{\frac{1}{p}}$ (**Norma di Frobenius**), $p \geq 1$

Teorema (Norma uniformemente continua):
La funzione $x \to \|x\|$ è uniformemente continua.

Teorema (Equivalenza fra norme vettoriali):
Per ogni coppia di norme vettoriali $\| \|$, $\| \|'$ su $\mathbb{C}^n$ esistono $a, b > 0$ tali che:

$a\|x\| \leq \|x\|' \leq b\|x\|$

Esempi utili:
$\|x\|_\infty \leq \|x\|_1 \leq n\|x\|_\infty$
$\|x\|_\infty \leq \|x\|_2 \leq \sqrt{n}\|x\|_\infty$

Definizione (Norma matriciale):

Una norma matriciale è un'applicazione $\| \|: M_n(\mathbb{C}) \to \mathbb{R}$ tale che:

$\|A\| \geq 0$; $\|A\| = 0 \leftrightarrow A = 0$

$\|aA\| = |a| \cdot \|A\|$

$\|A + B\| \leq \|A\| + \|B\|$

$\|AB\| \leq \|A\| \cdot \|B\|$ (Submoltiplicatività)

Esempio non indotta:

Norma di Frobenius: $\sqrt{tr(A^t A)}$ è invariante per trasformazioni unitarie.

Norma indotta dalla norma vettoriale:

Data un norma vettoriale $\| \|$ la norma matriciale indotta è:

$\|A\| \coloneqq \max_{\|x\|=1} \|Ax\|$

Le norme indotte misurano quanto A amplifica i vettori di norma unitaria.

Esempi:

$\|A\|_1 = \max_{j=1,\dots,n} \sum_{i=1}^{n} |a_{i,j}|$ (Massima colonna)

$\|A\|_2 = \sqrt{\rho(A^H A)}$

$\|A\|_\infty = \max_{i=1,\dots,n} \sum_{j=1}^{n} |a_{i,j}|$ (Massima riga)

Proprietà:

Se A è Hermitiana $\|A\|_2 = \rho(A)$

$\|\mathrm{Id}\| \geq 1$

Se la norma è indotta $\|\mathrm{Id}\| = 1$

Se A è invertibile $\|A^{-1}\| \geq \|A\|^{-1}$

$\left\|A^k\right\| \leq \|A\|^k$

Definizione (Raggio spettrale):

$\rho(A)$ è il massimo modulo degli autovalori della matrice.

Osservazione:

È una norma matriciale per le matrici Hermitiane.

$\forall$ $\| \|$ indotta vale $\|A\| \geq \rho(A)$

Teorema:

Per ogni $A \in M_n(\mathbb{C})$; $\forall\, \varepsilon > 0$ $\exists$ una norma matriciale indotta $\| \|$ tale che:

$\rho(A) \leq \|A\| \leq \rho(A) + \varepsilon$

Il raggio spettrale è dunque il limite inferiore di tutte le norme matriciali indotte.

Teorema fondamentale:

$\forall\, A \in M_n(\mathbb{C})$ e per ogni norma matriciale $\| \|$ vale:

$\lim_{k \to +\infty} \left\|A^k\right\|^{\frac{1}{k}} = \rho(A)$

Capitolo 4: Zeri di funzioni e problemi di punto fisso

Idea:
Per funzioni f semplici vogliamo individuare gli 0 (Ossia gli $x \mid f(x) = 0$) riconducendoci allo studio di problemi di punto fisso ($x \mid g(x) = x$).

Metodo:
Data $f(x)$ funzione di cui cerchiamo gli 0 possiamo studiare:
$g(x) \coloneqq x - \frac{f(x)}{h(x)}$ con $h(x)$ scelta a seconda del contesto.
Vale $g(a) = a \leftrightarrow f(a) = 0$

La successione che vogliamo converga al risultato è costruita nel seguente modo:
$$\begin{cases} x_0 \\ x_{k+1} = g(x_k) \end{cases}$$

Costruzione grafica:
Si rappresenta $g(x)$ su $\mathbb{R}^2$, i punti fissi corrispondono alle intersezioni di $g(x)$ con la bisettrice $y = x$.
Dato un punto x_k il modo di determinare il successivo è tracciare la retta verticale fino ad individuare l'intersezione con $g(x)$, proiettare il punto orizzontalmente sulla retta $y = x$ e usare questo punto per applicare nuovamente il metodo.

Condizione sufficiente di convergenza:
Sia a punto fisso per $g \in C^1([a-p, a+p])$ con $p > 0$ e tale che $|g'(x)| < 1 \; \forall \, x \in I = [a-p, a+p]$; sia inoltre $x_0 \in I$.
Allora $x_i \in I \; \forall \, i$ e $\lim x_i = a$

Teorema di unicità del punto fisso:
Se $g \in C^1([a,b])$ e $|g'(x)| < 1$ su $[a,b]$.
Allora esiste al più un punto fisso x per g in $[a,b]$

Teorema dell'errore :
Sia a un punto fisso per la funzione $g \in C^1(I)$; $\lambda \coloneqq \max\{|g'(x)| \mid x \in I\} < 1$; $I = [a-\rho, a+\rho]$ e $\sigma \coloneqq \frac{\delta}{1-\lambda}$ (**Intervallo di incertezza**) con δ maggiorazione dell'errore i-esimo.
Se $\sigma < \rho$ e $x_0 \in I$ allora:
$$|x_i - a| \leq \sigma + \lambda^i(\rho - \sigma)$$

Tipi di convergenza:

La convergenza si stima definendo:

$\gamma \coloneqq \lim_{k\to+\infty} \left|\frac{x_{k+1}-a}{x_k-a}\right|$

Oppure per funzioni derivabili osservando la derivata del punto di equilibrio a.

> **Attenzione:**
> È una stima del limite quindi non c'è alcuna garanzia che un metodo con convergenza lineare sia più lento di uno con convergenza superlineare su un numero basso di iterazioni,

Convergenza **geometrica** o **lineare**:

$0 < \gamma < 1 \sim 0 < |g'(a)| < 1$

Convergenza **Sublineare**:

$\gamma = 1 \sim |g'(a)| = 1$

Convergenza **Superlineare**:

$\gamma = 0 \sim |g'(a)| = 0$

> **Ordine di convergenza superlineare:**
> Definendo $\vartheta \coloneqq \lim_{k\to+\infty} \left|\frac{x_{k+1}-a}{(x_k-a)^p}\right|$ si dice che la convergenza è di ordine p se il limite esiste e appartiene a $]0, +\infty[$

Osservazione:
In realtà il calcolo di $g'(a)$ può essere evitato sfruttando il seguente teorema.

> **Teorema di convergenza superlineare:**
> Sia $g \in C^p([a,b])$; $g^{(1)}(\bar{x}) = \cdots = g^{p-1}(\bar{x}) = 0$ e $g^{(p)} \neq 0$ allora esiste un intervallo $I \coloneqq [\bar{x}-p, \bar{x}+p]$ allora $\forall x_0 \in I$ l successione converge ad $\bar{x}$ in modo superlineare con ordine p.
>
> **Viceversa:**
> Sia $g \in C^p([a,b])$; $g(\bar{x}) = \bar{x}$ ed esista un $x_0 \in [a,b]$ che converge in modo superlineare con ordine p a $\bar{x}$, allora $g^{(1)}(\bar{x}) = \cdots = g^{p-1}(\bar{x}) = 0$ e $g^{(p)} \neq 0$
>
> > **Attenzione:**
> > Questo metodo funziona per valori interi di p, quando così non è bisogna controllare a mano il limite per individuare l'ordine di convergenza.

Calcolo degli 0:

Metodo di bisezione:

Dato un intervallo $[a, b]$ | $f(a)f(b) < 0$ applicando il teorema degli 0 si calcola $c = \frac{1}{2}(b - a)$ e si valuta $f(c)$.
A seconda del segno si itera su $[a, c]$ o su $[c, b]$

Metodo di Newton:

Idea:
Dato un punto x_0 iniziale vicino ad a punti fisso si sceglie x_1 come l'intersezione della tangente con l'asse delle ascisse.

Formula:
$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$

Teorema convergenza metodo di Newton:
Se $f \in C^2([a, b])$ e $\bar{f}(x) = 0$ per un opportuno $\bar{x} \in [a, b]$; se $f'(\bar{x}) \neq 0$ allora esiste un intorno $I = [\bar{x} - p, \bar{x} + p] \subseteq [a, b]$ tale che $\forall\, x_0 \in I$ la successione converge ad $\bar{x}$ in maniera superlineare con ordine almeno 2.
L'ordine è esattamente 2 se $f''(x) \neq 0$

> **Osservazione:**
> Nel metodo di Newton più derivate di f si annullano più il punto sarà orizzontale e dunque il problema mal condizionato.

Condizioni di interruzione dell'iterazione:

Condizioni sul passo:

$|x_{k+1} - x_k| \leq \varepsilon$

Condizioni sul valore:

$|f(x_k)| \leq \varepsilon$

Problema degli 0 "piatti":
Se le derivate successive di f si annullano nello 0 $\bar{x}$, ossia:

$f(\bar{x}) = f'(\bar{x}) = \cdots f^{(p-1)}(\bar{x}) = 0\,; f^{(p)}(\bar{x}) \neq 0$

Definiamo $g(x) = x - \frac{f(x)}{f'(x)}$ imponendo $g(\bar{x}) = \bar{x}$.

La convergenza è allora del tipo $1 - \frac{1}{r}$

Capitolo 5: Trasformata discreta di Fourier (DFT)

Idea:
Definiremo le radici n-esime dell'unità e interpoleremo sfruttandone le proprietà.
In pratica useremo come nodi ω_n^j (Radici n-esime primitive), la matrice di Van der Monde associata sarà $F_{i,j} = \omega_n^{ij}$

Definizione (Radice n-esima dell'unità):
$\omega_n = \cos\left(\frac{2\pi}{n}\right) + i \cdot \sin\left(\frac{2\pi}{n}\right) = e^{\frac{2\pi i}{n}}$

Proposizione:
Data ω_n una radice dell'unità allora vale:

$$\sum_{j=0}^{n-1} \omega_n^{ij} = \begin{cases} n & \text{per } i \equiv 0\ (n) \\ 0 & \text{per } i \not\equiv 0\ (n) \end{cases}$$

Proposizione:
Sia F matrice di Van der Monde tale che $F_{i,j} = \omega_n^{ij}$, allora:

$F = F^t$
$F^H F = n \cdot \mathrm{Id}$

$$F^2 = n \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \iddots & \vdots \\ 0 & 1 & \cdots & 0 \end{pmatrix}$$

Corollario:
$F^{-1} = \frac{1}{n} F^H$
$\Omega \coloneqq \frac{1}{\sqrt{n}} F$ è unitaria e ha numero di condizionamento 1.

Definizione (Trasformata discreta di Fourier (DFT)):
$\mathrm{DFT}: \mathbb{C}^n \to \mathbb{C}^n \,;\, \mathrm{DFT}(y) \coloneqq F^{-1} y$

Definizione (Trasformata discreta inversa di Fourier (IDFT)):
$\mathrm{IDFT}: \mathbb{C}^n \to \mathbb{C}^n \,;\, \mathrm{IDFT}(x) \coloneqq Fx$

Osservazione:
Calcolare la DFT corrisponde a calcolare i coefficienti del polinomio interpolante.
Calcolare la IDFT corrisponde a valutare il polinomio interpolante.

Algoritmi (FFT):

Algoritmo di Cooley – Turkey (IDFT):

Vogliamo determinare gli y_k dati i punti x_k, per semplicità supponiamo $n = 2^a$; $m = \frac{n}{2}$

Il procedimento prevede di spostare il calcolo della IDFT allo studio di due di dimensione minore (Dimezzata):

$$y_k = \sum_{j=0}^{m-1} \omega_m^{jk} \cdot x_{2j} + \omega_n^k \sum_{j=0}^{m-1} \omega_m^{jk} \cdot x_{2j+1}$$

Dunque scrivendo $y_k = y_{m+i}$ per le seconde componenti vale:

$$y_k = y_{m+i} = \sum_{j=0}^{m-1} \omega_m^{j(m+i)} \cdot x_{2j} + \omega_n^{m+i} \sum_{j=0}^{m-1} \omega_m^{j(m+i)} \cdot x_{2j+1}$$

Ma $\omega_m^{m+i} = \omega_m^i$ quindi abbiamo già calcolato questi valori mentre $\omega_n^{m+i} = -\omega_n^i$ bisogna considerare le sottrazioni.

Costo computazionale:

Applicando questo procedimento vale la formula ricorsiva:

$$\begin{cases} \text{Costo}_n = 2 \cdot \text{Costo}_{\frac{n}{2}} + \frac{n}{2} M + n \cdot A \\ \text{Costo}_1 = 0 \end{cases}$$

Quindi:

$$\text{Costo}_n = \log_2 n \left(\frac{n}{2} M + nA \right) \sim \frac{3}{2} n \log_2 n$$

Applicazioni trasformate di Fourier:

Oltre a determinare i coefficienti di un polinomio interpolante la DFT può essere usata anche per:

Moltiplicazione di polinomi:

Dati:

$$\begin{cases} a(x) = \sum_{i=0}^{n_a} a_i \cdot x^i \\ b(x) = \sum_{i=0}^{n_b} b_i \cdot x^i \\ c(x) = a(x)b(x) = \sum_{i=0}^{n_a+n_b} c_i \cdot x^i \end{cases}$$

Supponendo $n_a \geq n_b$ possiamo calcolare i coefficienti di c mediante la moltiplicazione matrice vettore:

$$Ab = \begin{pmatrix} a_0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ a_{n_a} & \cdots & \cdots & a_0 \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & a_{n_a} \end{pmatrix} \begin{pmatrix} b_0 \\ \vdots \\ \vdots \\ b_{n_b} \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Di base questo ha un costo di $2n_a n_b \sim O(n^2)$

Applicazione (IDFT e DFT):

Sia $N \coloneqq 2^a \geq n_a + n_b + 1$, consideriamo la IDFT per valutare i polinomi sulle radici N-esime dell'unità, ossia:

$a_i \coloneqq a\left(\omega_N^i\right) ; b_i = b\left(\omega_N^i\right)$ con costo $2\left(\frac{3}{2} N \log_2 N\right)$

Si ricavano:

$\gamma_i \coloneqq a_i b_i = c\left(\omega_N^i\right)$ con costo N

Infine si applica la DFT a $(\gamma_0 \quad \cdots \quad \gamma_{N-1})$ con costo $\frac{3}{2} N \log_2 N$

Costo totale: $\frac{9}{2} N \log_2 N + N \sim O(N \log_2 N)$

Interpolazione trigonometrica:

Un'interpolazione trigonometrica è data da un polinomio della forma:

$$\frac{a_0}{2} + \sum_{j=1}^{m-1}\left(a_j \cdot \cos(jx) + b_j \cdot \sin(jx)\right) + \frac{a_m}{2}\cos(mx)$$

Per individuare i coefficienti a_i ; b_j dati i nodi $x_k \coloneqq \frac{2\pi k}{n}$; $y_k = p(x_k)$

Applicazione DFT:

Se calcoliamo con gli algoritmi noti $z_j = DFT\left(y_j\right)$ allora:

$$\begin{cases} a_j = 2 \cdot \mathrm{Re}\left(z_j\right) \\ b_j = -2 \cdot \mathrm{Im}\left(z_j\right) \end{cases}$$

Osservazione:

Esiste un unico polinomio trigonometrico con $m = n$ che risolve il problema di interpolazione.

Capitolo 6: Interpolazione

Idea:
Data una funzione $f(x)$ definita su di un intervallo $[a, b]$, noti $n+1$ punti del grafico detti nodi $\{(x_0, y_0), \dots, (x_n, y_n)\}$ e un insieme di funzioni $F = \{\varphi_j(x)\ ; j = 0, \dots, n\}$ definite su $[a, b]$ e linearmente indipendenti, interpolare significa individuare la funzione:

$g(x) = \sum_{j=0}^{n} a_j \cdot \varphi_j(x)$

Tale che $g(x_i) = f(x_i) = y_i \ \ i = 0, \dots, n+1$
La scelta di funzioni facili permette di lavorare con oggetti più maneggevoli.

Osservazione:
La soluzione se esiste dipende dalla scelta della base.

A seconda di questa scelta di funzioni interpolanti si parla di:

- Interpolazione razionale (Funzioni razionali)
- Interpolazione polinomiale (Polinomi)
- Interpolazione trigonometrica (Seni e coseni)

Interpolazione polinomiale:

La scelta di una base per l'interpolazione polinomiale è $\{1, x, \dots x^n\}$ e la matrice associata al problema risulta essere una matrice di Van der Monde.

Matrici di Van der Monde:

Sia $V_n = \begin{pmatrix} 1 & x_0 & \cdots & x_0^n \\ 1 & x_1 & \cdots & x_1^n \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \cdots & x_n^n \end{pmatrix}$ allora $\det V_n = \prod_{i>j}(x_i - x_j)$

Osservazione:

Il numero di condizionamento di una matrice di Van der Monde cresce esponenzialmente con n.
Se gli x_i vengono scelti in alcuni modi si può abbassare il numero di condizionamento (Ad esempio se scegliamo le radici n-esime dell'unità come nella DFT).

Costo computazionale:

Per risolvere un sistema di interpolazione abbiamo il costo usuale di $\frac{2}{3}n^3$.
Vista la struttura di queste matrici esistono due algoritmi di costo rispettivamente:

n^2

$n \log^2 n$ con problemi di instabilità

Il prodotto matrice vettore ha un costo di n^2 oppure $n \log n$ con problemi di instabilità.

Resto dell'interpolazione:

Definito il resto dell'interpolazione come $r_n(x) = f(x) - p_n(x)$ vale il seguente teorema.

Teorema del resto:

Data $f \in C^{n+1}[a, b]$ allora preso $x \in [a, b]$ esiste $\varepsilon \in [a, b]$ tale che:

$r_n(x) = \prod_{i=0}^{n}(x - x_i)\frac{f^{(n+1)}(\varepsilon)}{(n+1)!}$

Osservazione:

Nel caso dell'interpolazione polinomiale questa maggiorazione è poco stringente.

Osservazione (DFT):

Come visto prima individuare la DFT è equivalente a calcolare i coefficienti del polinomio interpolante, la strategia è dunque ricondurci al caso particolare del capitolo precedente e applicare gli algoritmi visti.

Interpolazione polinomiale di Lagrange:

Idea:
Risolvere i problemi di mal condizionamento delle matrici di Van der Monde.
Consideriamo la famiglia di polinomi:

$$L_i(x) \coloneqq \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x-x_j}{x_i-x_j}$$

Osservazione:
La famiglia sopra scritta è una base (Detta base di Lagrange) per i polinomi fino al grado n-esimo.

Polinomio di interpolazione:
Siccome in questa base la matrice di Van der Monde diventa l'identità allora il polinomio interpolante è semplicemente:

$$p(x) = \sum_{i=0}^{n} y_i \cdot L_i(x)$$

Valutazione del polinomio interpolante in un punto z:

$$p(z) = \prod_{k=0}^{n}(z - x_k) \sum_{i=0}^{n} \frac{y_i}{(z-x_i)\prod_{j\neq i}(x_i-x_j)}$$

Osservazione costo computazionale:
La produttoria al denominatore costa $O(n^2)$ ma può essere calcolata una volta sola.
Una volta fatto il calcolo di $p(z)$ ha costo lineare in n.

Capitolo 7: Sistemi lineari

Idea:
Studiare un sistema lineare significa risolvere il problema:

$Ax = b$ con $A \in M_n(\mathbb{C})$; $b \in \mathbb{C}^n$

Un sistema è detto **consistente** se ammette almeno una soluzione non nulla.

> **Osservazione:**
> In questo caso potremmo calcolare A^{-1} e risolverlo imponendo $x = A^{-1}b$ ma come sistema non è ottimale.

Metodi diretti e iterativi:
I metodi diretti fattorizzano A così da rendere semplice il calcolo.
I metodi iterativi individuano una successione di vettori convergenti alla soluzione del sistema.

Condizionamento (Numero di condizionamento):

Quando si parla di condizionamento intendiamo studiare come vari la soluzione in funzione di perturbazione dei dati iniziali.

Perturbazione del vettore b dei termini noti:

Al posto di b consideriamo $b + \delta_b$. Il problema diventa:

$A(x + \delta_x) = b + \delta_b$ e vogliamo studiare δ_x

Al posto di un'analisi dell'errore usiamo le norme per stimare l'errore relativo.

$$\frac{\|\delta_x\|}{\|x\|} \leq \|A\| \cdot \|A^{-1}\| \cdot \frac{\|\delta_b\|}{\|b\|}$$

Definizione (Numero di condizionamento):

Definiamo il **numero di condizionamento** della matrice A come il coefficiente di amplificazione individuato:

$$K(A) \coloneqq \|A\| \cdot \|A^{-1}\|$$

Osservazione:

Se stiamo applicando più cambi di basi $K(VS) < K(V)K(S)$

Quindi dobbiamo cercare di effettuare dei cambi di base unitari.

Perturbazione della matrice A:

In questo caso generale stiamo studiando il sistema:

$$(A + \delta_A)(x + \delta_x) = b + \delta_b$$

Se la perturbazione è piccola rispetto alla matrice $\|\delta_A\| \cdot \|A^{-1}\| < 1$ ci si riconduce al caso precedente.

In generale vale:

$$\frac{\|\delta_x\|}{\|x\|} \leq \frac{K(A)}{1 - \|\delta_A\| \cdot \|A^{-1}\|} \cdot \left(\frac{\|\delta_b\|}{\|b\|} + \frac{\|\delta_A\|}{\|A\|}\right)$$

Casi noti (In norma 2):

Matrice unitaria: $K(Q) = 1$

Matrice Hermitiana: $K(H) = \frac{\lambda_{\max}}{\lambda_{\min}}$

Metodi diretti per sistemi lineari:

Idea:
Ricondurci ad un caso nel quale è semplice risolvere il problema, ad esempio matrici triangolari o unitarie.
Se stiamo cercando di individuare la soluzione del sistema $Ax = b$ e $A = BC$ possiamo scrivere il problema come:

$$BCx = b \rightarrow \begin{cases} By = b \\ Cx = y \end{cases}$$

Osservazione tridiagonali:

Se ci troviamo nel caso in cui dobbiamo risolvere $Ax = b$ con A **triangolare superiore** allora mediante il metodo di sostituzione all'indietro la soluzione è:

$$\begin{cases} x_n = \frac{b_n}{a_{n,n}} \\ x_{n-i} = \frac{\left(b_{n-i} - \sum_{j=i+1}^{n} a_{n-i,j} x_j\right)}{a_{n-i,n-i}} \end{cases}$$

Se **triangolare inferiore** la sostituzione in avanti è invece:

$$\begin{cases} x_1 = \frac{b_1}{a_{1,1}} \\ x_i = \frac{\left(b_i - \sum_{j=1}^{i-1} a_{i,j} x_j\right)}{a_{i,i}} \end{cases}$$

Osservazione:
Il costo è di $O(n^2)$ e i metodi sono stabili all'indietro.

Osservazioni unitarie:

Se abbiamo $Ax = b$ con A unitaria allora semplicemente:
$x = A^H b$

Osservazione:
Il costo è di $2n^2 - n \sim O(n^2)$ e il metodo è stabile all'indietro.

Matrici elementari:

Una matrice si dice elementare se è della forma $E \coloneqq \mathrm{Id} - \sigma u v^H$ con $\sigma \in \mathbb{C}$; $u, v \in \mathbb{C}^n$
Possono essere sfruttare per determinare delle decomposizioni in forma facile di matrici non particolari.

Osservazione rango:
$\mathrm{rango}(uv^H) = 1$

Osservazione autovalori:
Gli autovalori di una matrice elementare sono:

1 con molteplicità $n - 1$

$(1 - \sigma v^H u)$

Inoltre E è invertibile $\leftrightarrow\ \sigma v^H u \neq 1$

Osservazione inversa:
L'inversa di una matrice elementare è elementare.
Se $E = \mathrm{Id} - \sigma u v^H$ allora $E^{-1} = \mathrm{Id} - \tau u v^H$ con $\tau = \frac{\sigma}{\sigma(v^H u) - 1}$

Il costo per calcolare l'inversa è lineare $\big(O(n)\big)$

Proprietà vettori:
Dati $x, y \in \mathbb{C}^n$ non nulli allora esiste una matrice elementare non singolare E tale che:

$Ex = y$

Conseguenza:
Possiamo ogni volta applicare l'osservazione precedente per assegnare il j-esimo elemento della matrice A in un vettore a nostra scelta.

Dunque $A = E_1^{-1} \dots E_{n-1}^{-1} A_n$ con A_n triangolare superiore fino alla riga n-esima.

Householder e Gauss:
Se scegliamo le matrici elementari unitarie (Householder) otteniamo una fattorizzazione QR.
Se scegliamo le matrici elementari triangolari inferiori (Gauss) otteniamo una fattorizzazione LU.

Fattorizzazione LU:

Idea:

$A = LU$ con L triangolare inferiore con diagonale uguale ad 1 e U triangolare superiore.

Si può fattorizzare in questo modo una matrice complessa se e solo se tutte le sottomatrici principali di testa sono invertibili.

L'algoritmo utilizzato per il calcolo della fattorizzazione LU è l'eliminazione gaussiana.

Senza applicar strategie di pivoting il metodo è instabile dal punto di vista numerico.

Teorema di esistenza e unicità:

$A \in M_n(\mathbb{C})$ ammette un'unica fattorizzazione LU se e solo se tutte le sottomatrici principali di testa di dimensione minore uguale ad $n - 1$ sono invertibili.

Matrici di Gauss:

Matrici elementari tali che $v = e_1$; $u_1 = 0$.

$$E = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ -u_1 & & & \\ \vdots & & \text{Id} & \\ -u_n & & & \end{pmatrix}$$

Osservazione:

Si sfrutta il fatto che $Ex = \begin{pmatrix} x_1 \\ x_2 - u_2 x_1 \\ \vdots \\ x_n - u_n x_1 \end{pmatrix}$

Metodo diretto per calcolare i coefficienti:

Data la matrice:

$$A_k = \begin{pmatrix} a_{1,1} & \cdots & a_{1,k} & a_{1,k+1} & \cdots & a_{1,n} \\ 0 & \ddots & \vdots & \vdots & & \vdots \\ \vdots & 0 & a_{k,k} & a_{k,k+1} & \dots & a_{k,n} \\ \vdots & \vdots & a_{k+1,k} & a_{k+1,k+1} & \cdots & a_{k+1,n} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & a_{n,k} & a_{n,k+1} & \cdots & a_{n,n} \end{pmatrix}$$

Determinare la matrice elementare di Gauss:

E_k è la matrice identità con la colonna k sostituita dal vettore:

$$\begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ -m_{k+1,k} \\ \vdots \\ -m_{n,k} \end{pmatrix}$$ con $m_{i,k} = a_{i,k} \cdot a_{k,k}^{-1}$ con $a_{i,k}$; $a_{k,k}$ coefficienti della matrice A_k

Ricaviamo dunque:

$$a_{i,j}^{(k+1)} = \begin{cases} 0 & \text{se } i = k, j > k \\ a_{i,j}^{(k)} - m_{i,k} \cdot a_{k,j}^{(k)} & \text{se } i, j > k \\ a_{i,j}^{(k)} & \text{altrove} \end{cases}$$

Una volta ottenuta $A_n = E_{n-1} \cdots E_1 A$ allora:

$U = A_n$

$L = E_1^{-1} \cdots E_{n-1}^{-1}$

Osservazione inversa:

E_k^{-1} essendo elementare si ottiene cambiando di segno gli elementi non sulla diagonale $(m_{i,k} \to -m_{i,k})$

Il prodotto di tutte queste si ottiene semplicemente come la matrice Id con nelle sottodiagonali i valori calcolati subito prima.

Costo dell'eliminazione gaussiana:
Individuare la matrice di Gauss necessita di $(n-k)$ operazioni
Il calcolo della matrice A_{k+1} costa invece $2(n-k)^2$.
Dunque:

$$C_{tot} = \frac{2}{3}n^3 + O(n^2)$$

Casi particolari:
Per le matrici a banda il costo è di $O(n^2)$
Per le matrici di Hessenberg il costo è di $O(n^2)$

Memoria occupata:
La memoria occupata si riduce ad n^2 se usiamo il seguente accorgimento:
A_{k+1} va a sovrascrivere i valori di A_k e negli spazi vuoti al di sotto della diagonale scriviamo i valori ricavati delle matrici di Gauss.

Instabilità:
Stando dividendo per $a_{k,k}$ non abbiamo garanzie che il metodo sia stabile.

Pivoting LU (PLU):
L'idea è che quando l'algoritmo incontro uno 0 in uno dei posti $a_{k,k}$ va ad applicare una matrice di permutazione così da eliminarlo e poter proseguire con l'algoritmo.

Osservazione:
A meno di permutazioni esiste sempre una fattorizzazione PLU.

Strategia del massimo pivot parziale:
Permutiamo in modo da portare nella posizione k, k l'elemento di modulo massimo lungo la colonna k.

Strategia del massimo pivot totale:
Permutiamo in modo da portare nella posizione k, k l'elemento di modulo massimo dell'intera sottomatrice di coda.

Fattorizzazione QR:

Idea:
$A = QR$ con Q unitaria e R triangolare superiore.
A differenza della fattorizzazione LU esiste sempre una fattorizzazione QR della matrice ma non è unica.
Il costo computazionale è doppio ma questo metodo è stabile all'indietro.

Teorema di esistenza:
Data una matrice A esiste sempre la fattorizzazione QR ma questa non è unica.

> **Osservazione:**
> A meno di trasformazioni unitarie diagonali diventa unica.

Matrici di Householder:
Sono matrici elementari della forma:

$$E = \mathrm{Id} - buu^H$$

Metodo diretto per calcolare i coefficienti:

Data la matrice A_k come prima, allora la matrice elementare di Householder è:

$$M_k = \mathrm{Id} - b_k u^{(k)} u^{(k)^H}$$

Ottenibile come:

$$u_i^{(k)} = \begin{cases} 0 & \text{se } i < k \\ a_{k,k}^{(k)} \left(1 + \frac{\left(\sum_{r=k}^{n} \left| a_{r,k}^{(k)} \right|^2 \right)^{\frac{1}{2}}}{\left| a_{k,k}^{(k)} \right|} \right) & \text{se } i = k \\ a_{i,k}^{(k)} & \text{se } i > k \end{cases}$$

E la matrice successiva invece:

$$a_{i,j}^{(k+1)} = \begin{cases} a_{i,j}^{(k)} & \text{se j} < k \text{ ; } i \leq k \\ 0 & \text{se j} = \text{k ; } i > k \\ a_{i,j}^{(k)} - b^{(k)} u_i^{(k)} \sum_{r=k}^{n} u_r^{(k)} b_r^{(k)} & \text{se j} > k \text{ ; } i \geq k \end{cases}$$

Costo computazionale:

L'algoritmo sopra descritto ha un costo di $\frac{4}{3}n^3 + O(n^2)$

Questo algoritmo dunque ha un costo doppio rispetto a quello della fattorizzazione LU.

Stabilità QR:

La fattorizzazione QR è stabile per analisi all'indietro dell'errore.

Metodi iterativi per sistemi lineari:

Idea:
Un metodo iterativo diventa efficace se sappiamo calcolare a costo basso il prodotto matrice vettore, ad esempio quando A è una matrice sparsa.

Metodo:
Dato il sistema $Ax = b$ si scrive $A = M - N$ con M facilmente invertibile.
Allora vale:

$$Mx = Nx + b \rightarrow x = M^{-1}Nx + M^{-1}b$$

Il problema diventa quindi la ricerca di un punto fisso per la funzione prima descritta.

$P \coloneqq M^{-1}N$ è detta **matrice di iterazione**, $q \coloneqq M^{-1}b$

> **Osservazione:**
> Se la matrice di iterazione ha raggio spettrale nullo il metodo diviene diretto e la successione coinciderà definitivamente con la soluzione.

La successione da studiare è:

$$\begin{cases} x_0 \text{ qualsiasi} \\ x_{k+1} = Px_k + q \end{cases}$$

Convergenza:

Definizione (Successione di vettori convergenti):
Una successione di vettori $x_k \in \mathbb{C}^n$ converge ad un vettore $\bar{x}$ se esiste una norma vettoriale per cui:
$\lim_{k\to\infty} \|x_k - \bar{x}\| = 0$

> **Osservazione:**
> Non dipende dalla scelta della norma ma di solito lavoriamo in norma 2.

> **Definizione (Metodo iterativo convergente):**
> Un metodo iterativo si dice convergente se $\forall\, x_0 \in \mathbb{C}^n$ vale $\lim x_k = \bar{x}$ soluzione del sistema.

> **Definizione (Errore al passo k):**
> $e_k \coloneqq x_k - \bar{x}$

Condizione sufficiente di convergenza:
Se esiste una norma matriciale indotta tale che $\|P\| < 1$ allora la successione converge.

> **Osservazione:**
> Controllare come prima cosa la norma 1 e la norma ∞.

<u>Condizione necessaria e sufficiente di convergenza:</u>
$\forall\, x_0\,, \lim x_k = \bar{x} \leftrightarrow \rho(P) < 1$

Confronto fra matrici di iterazione:
Per confrontare due metodi iterativi non possiamo valutare le norme delle matrici poiché il risultato dipenderebbe dalla scelta della norma.

> **Osservazione:**
> Possiamo maggiorare l'errore al passo k-esimo con il raggio spettrale.
>
> **Dunque:**
> Se confrontiamo però il raggio spettrale possiamo determinare quale metodo converge più rapidamente.
>
> > **Attenzione:**
> > È una maggiorazione relativa al caso peggiore, se sappiamo escludere gli autospazi con modulo degli autovalori maggiori il risultato potrebbe cambiare.

Osservazione matrici singolari:
Se A è singolare il metodo non converge (Almeno lungo gli autospazi relativi a 0)

Condizioni di arresto:

Valutazione del residuo:

$\|\bar{x} - x_k\| \leq \|A^{-1}\| \cdot \|r_k\| \leq \varepsilon \cdot \|A^{-1}\|$

Con $r_k \coloneqq b - Ax_k$ il residuo.

L'errore relativo si maggiora con:

$$\frac{\|\bar{x}-x_k\|}{\|x\|} \leq \varepsilon \frac{K(A)}{\|b\|}$$

Valutazione del passo:

Dalla relazione:

$x_k - x_{k-1} = (\mathrm{Id} - P)(\bar{x} - x_{k-1})$

Passiamo alle norme e alla valutazione dell'errore sfruttando la maggiorazione:

$$\|(\mathrm{Id} - P)^{-1}\| \leq (1 - \|P\|)^{-1}$$

Dunque:

$$\|x_k - x_{k-1}\| \leq (1 - \|P\|)^{-1}\varepsilon$$

Se $\|P\|$ è vicino a 0 converge rapidamente, al contrario se $\|P\|$ è vicino ad 1 allora la convergenza è lenta.

Metodo di Jacobi:

$A = D - B - C$ con:

D diagonale non nulla

B triangolare inferiore

C triangolare superiore

Scelta di M, N, P:

$M = D$

$N = B + C$

$P = D^{-1}(B + C) \coloneqq J$

Iterazione:

$x_{k+1} = D^{-1}\big((B + C)x_k + b\big)$

Iterazione componente a componente:

$$x_{k+1,i} = \frac{1}{a_{i,i}}\left(-\sum_{\substack{j=0 \\ j\neq i}}^{n} a_{i,j}x_{k,j} + b_i\right)$$

Osservazioni:

Il costo è dato prevalentemente dal prodotto $(B + C)x_k$

Dobbiamo mantenere in memoria due vettori.

Metodo implementabile in parallelo componente a componente.

Metodo di Gauss – Seidel:

$A = D - B - C$ con:

D diagonale non nulla

B triangolare inferiore

C triangolare superiore

Scelta di M, N, P:

$M = D - B$

$N = C$

$P = (D - B)^{-1}C \coloneqq G$

Iterazione:

$x_{k+1} = D^{-1}(Bx_{k+1} + Cx_k + b)$

Iterazione componente a componente:

$x_{k+1,i} = \frac{1}{a_{i,i}}\left(-\sum_{j=0}^{i-1} a_{i,j}x_{k+1,j} - \sum_{j=i+1}^{n} a_{i,j}x_{k,j} + b_i\right)$

Osservazioni:

Questo metodo è in media più veloce ma non è implementabile in parallelo.

Dobbiamo mantenere in memoria un solo vettore.

Teoremi di convergenza:

Ricordando che se il raggio spettrale della matrice è minore di 1 i metodi convergono valgono i seguenti risultati.

Teorema di dominanza:

Se A soddisfa una delle seguenti condizioni allora $\rho(J) < 1$; $\rho(G) < 1$:

A è fortemente dominante diagonale.
A è dominante diagonale e irriducibile.
A^t è fortemente dominante diagonale.
A^t è dominante diagonale e irriducibile.

Teorema delle tridiagonali:

Se A è tridiagonale con elementi sulla diagonale diversi da 0 allora $\rho(G) = \rho(J)^2$

Teorema di Stein – Rosenberg:

Se A è tale che $a_{i,i} > 0 \; \forall \; i$ e $a_{i,j} \leq 0$ allora si verifica uno e uno solo dei seguenti risultati:

$\rho(G) = \rho(J) = 0$
$\rho(G) = \rho(J) = 1$
$\rho(G) < \rho(J) < 1$ (Se Jacobi converge allora Gauss – Seidel converge più velocemente)
$1 < \rho(J) < \rho(G)$

Capitolo 8: Metodi di rilassamento (SOR) e del gradiente

Idea:
Per risolvere i sistemi lineari conosciamo già i metodi diretti e i metodi iterativi di Jacobi e Gauss Seidel
Per casi troppo grandi o lenti nella convergenza dei metodi classici introduciamo questi nuovi algoritmi.

I metodi del gradiente sfruttano proprietà della matrice delle derivate successive per determinare una direzione di decrescita.

Il metodo del rilassamento (SOR) è invece una variazione del metodo di Gauss – Seidel.

Metodo del gradiente:

Data $A \in M_n(\mathbb{R})$ definita positiva e il sistema lineare che vogliamo studiare:

$$Ax = b$$

Se il costo del prodotto matrice vettore è basso possiamo usare un metodo iterativo.

Considerando:

$$\Phi(x) = \frac{1}{2}x^t Ax - x^t b$$

Il suo gradiente vale:

$$\nabla\Phi(x) = Ax - b$$

Osservazione:

$\nabla\Phi(x) = 0 \leftrightarrow x$ risolve il sistema lineare. (E

Riduciamo il problema alla ricerca dei punti stazionari di $\Phi(x)$

Punti di minimo:

La matrice delle derivate seconde di $\Phi(x)$ è A (definita positiva) nei punti stazionari, dunque questi sono dei punti di minimo.

$A\hat{x} = b \leftrightarrow \hat{x} = \min_{x\in\mathbb{R}^n}\big(\Phi(x)\big)$

Procedimento:

Scegliamo x_0 arbitrario.

Si determina la direzione del minimo.

Si pone $x_1 = x_0 + a_0 v_0$ con v_0 vettore di decrescita e a_0 uno scalare opportuno.

Osservazione:

La determinazione della direzione del minimo può essere diversa e origina diversi metodi.

Metodo del gradiente ottimo:

Scegliamo la direzione di decrescita come l'opposto della direzione di massima crescita indicata dal gradiente:

$v_k = -\nabla\Phi(x)$

Con $a_k = \frac{v_k^t r_k}{v_k^t A v_k}$ con $r_k = b - Ax_k$

Motivazione:
Per la scelta di a_k valutiamo $\Phi(x)$ lungo la retta determinata da v_k.

$$g(a) = \Phi(x_k + av_k)$$

Cerchiamone il minimo osservando che è convessa (Quindi basta trovare il punto stazionario):

$$g'(a) = av_k^t A v_k + v_k^t(Ax_k - b) = 0$$

Da cui $a_k = \frac{v_k^t r_k}{v_k^t A v_k}$ con $r_k = b - Ax_k$

Proprietà:
Convergente per ogni scelta di x_0.

$\|r_k\| = \|Ax_k - b\|$ è indice di quanto siamo lontani dalla soluzione, ci fornisce quindi una condizione di arresto..

Osservazione computazionale:
Potremmo evitarci il calcolo di $r_{k+1} = b - Ax_{k+1} = r_k - a_k A v_k$

Attenzione:
Riduce il costo ma standolo ricavando iterativamente amplifica l'errore proprio nel termine che usiamo per decidere quanto interrompere l'iterazione.

Metodo del gradiente coniugato:

Osservazione:
Questo metodo in realtà se consideriamo n iterazioni diventa diretto.

Definizione A-coniugati:
Data una matrice $A \in M_n(\mathbb{R})$ definita positiva e simmetrica ed una n-upla di vettori $(p_1, \dots, p_n)$ di $\mathbb{R}^n$ si dicono A-coniugati se:

$$\begin{pmatrix} p_1^t \\ \vdots \\ p_n^t \end{pmatrix} A(p_1 \quad \cdots \quad p_n) = \text{Diagonale}$$

Osservazione:
Se A è invertibile allora questi vettori sono indipendenti.
Quindi non possiamo fare più di n iterazioni mantenendo questo tipo di vettori.

Costruzione dei vettori A-coniugati:
Per costruire una successione di vettori che siano A-coniugati poniamo:

$$v_k = r_k + B_k v_{k-1}$$

Con $B_k = -\frac{r_k^t A v_{k-1}}{v_{k-1}^t A v_{k-1}}$

Costruzione della successione x_k:
Invece gli x_k vengono definiti con la stessa regola del gradiente imponendo come velocità di decrescita proprio v_k (Mentre a_k viene individuato come prima).

Teorema convergenza:
Siano $\{x_k\}_{k=1,\dots,n}$ una succssione di vettori costruiti con la regola di prima e con $x_0 = 0$, allora $\forall\ k$ vale:

$$\Phi(x_k) = \min_{x \in <v_1,\dots,v_{k-1}>}\big(\Phi(x)\big)$$

Quindi $\Phi(x_n)$ è il minimo assoluto.

Motivazione metodo iterativo:
Sebbene il teorema precedente affermi che in realtà il metodo sia diretto possiamo voler eseguire un numero più basso di iterazioni considerandolo iterativo.

Teorema errore:
Per ogni scelta iniziale x_0 vale:

$$\|e_k\|_A \leq \left(\frac{2\sqrt{K(A)}-1}{\sqrt{K(A)}+1}\right)^k \|e_0\|_A$$

Quindi se la matrice A è ben condizionata il metodo converge molto velocemente.

Precondizionamento:

Idea:
Per il teorema sull'errore una matrice ben condizionata converge rapidamente.

Nel caso di una matrice mal condizionata possiamo dunque portarci al caso di una ben condizionata così da incrementare la velocità di convergenza.

> **Attenzione:**
> Partendo da una matrice A mal condizionata non abbiamo garanzie sulla precisione del risultato.

Metodo:
Dato il sistema $Ax = b$ consideriamo:

$$LAL^t(L^t)^{-1}x = Lb$$

Inoltre vale la relazione di similitudine $LAL^t \sim L^{-1}LAL^tL \sim AL^tL \coloneqq AM$
Studiamo come far variare M.

> **Osservazione inversa:**
> M migliore sarebbe A^{-1}, se lo conoscessimo non si porrebbe il problema quindi in realtà ricercheremo delle approssimazioni di A^{-1}.

> **Osservazione condizionamento:**
> Il condizionamento della matrice è maggiorato da $\frac{\lambda_{max}}{\lambda_{min}}$.

Quindi cerchiamo M tale che AM abbia gli autovalori vicini.

Problemi:
M sembrerebbe dover essere definita positiva (Esiste un metodo per risolvere questo problema).
Dopo aver calcolato M apparente dovremmo ricavare L ma in realtà possiamo farne a meno.
Il modo di precondizionare dipende caso per caso.

Metodi di rilassamento (SOR Successive Over Relaxation):

A partire dal metodo di Gauss – Seidel la cui iterazione può essere scritta nella forma:

$$x^{(k)} = x^{(k-1)} + r^{(k)}$$

Con $r^{(k)} = x^{(k)} - x^{(k-1)} = D^{-1}\left(Bx^{(k)} + Cx^{(k-1)} + b\right) - x^{(k-1)}$ il residuo.

Allora conviene applicare un correttivo definendo:

$x^{(k)} = x^{(k-1)} + \omega \cdot r^{(k)}$

Definizioni:

Se $\omega < 1$ si parla di **sottorilassamento**

Se $\omega = 1$ è il normale metodo di Gauss – Seidel

Se $\omega > 1$ si parla di **sovrarilassamento**

Metodo:

$x^{(k)} = H(\omega)x^{(k-1)} + \omega(D - \omega B)^{-1}b$

Con $H(\omega) = (D - \omega B)^{-1}\left((1-\omega)D + \omega C\right)$ matrice di iterazione.

Componenti:

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

Osservazione:

Dobbiamo scegliere ω così da ridurre il raggio spettrale di $H(\omega)$ e aumentare la velocità di convergenza.

Teorema di Kahan:

Per la matrice di iterazione di un metodo di rilassamento risulta:

$$\rho\left(H(\omega)\right) \geq |\omega - 1|$$

Quindi condizione necessaria per la convergenza:

$$|\omega - 1| < 1$$

Caso reale:

$$0 < \omega < 2$$

Teorema di Ostrawski – Reich:

Se A è definita positiva e ω è un numero reale tale che $0 < \omega < 2$ allora il metodo di rilassamento è convergente.

Teorema di minimizzazione:

Sia A una matrice tridiagonali a blocchi (Blocchi quadrati e non singolari) e $0 < \omega < 2$.

Dette:

J_B matrice di iterazione del metodo di Jacobi a blocchi.

$H_B(\omega)$ matrice di iterazione del metodo di rilassamento a blocchi.

Allora $\forall\ \mu$ autovalore di J_B e $\lambda \mid (\lambda + \omega - 1)^2 = \lambda\omega^2\mu^2$ è autovalore di $H_B(\omega)$

(Valido anche il viceversa)

Inoltre se gli autovalori di J_B sono reali e $\rho(J_B) < 1$ il valore ottimo ω_0 del metodo di rilassamento a blocchi è:

$$\omega_0 = \frac{2}{1+\sqrt{1-\rho^2(J_B)}}$$

Infine:

$$\rho\big(H_B(\omega_0)\big) = \omega_0 - 1 = \left(\frac{\rho(J_B)}{1+\sqrt{1-\rho^2(J_B)}}\right)^2$$

Pratica:

Se studiamo la matrice di iterazione di Jacobi J possiamo ricavare delle condizioni per minimizzare la matrice di iterazione del SOR.

Capitolo 11: Problema lineare dei minimi quadrati

Idea:
Aggiungere ai metodi già studiati per risolvere i problemi lineari una diversa formulazione che permette di estenderli alle matrici rettangolari.
Dal punto di vista pratico questo significa studiare problemi di interpolazione dove il polinomio interpolante ha grado strettamente minore dei punti dei quali possediamo informazioni.

Problema:
Siano $A \in M_{m\times n}(\mathbb{R})$; $b \in \mathbb{R}^m$.
La ricerca di x che risolve:

$$Ax = b$$

Presenta due problemi, la difficoltà della risoluzione del sistema lineare e l'effettiva impossibilità a risolverlo se $b \notin \mathrm{Imm}(A)$.

Problema ai minimi quadrati equivalente:
Cerchiamo:
$x = \min_{x\in\mathbb{R}^n}(\|Ax - b\|_2)$
Così da avere la migliore approssimazione del problema.

> **Osservazione interpolazione:**
> Avendo posto il problema in questa forma A non deve più necessariamente essere quadrata.
> Questa formulazione può dunque risolvere il problema dell'interpolazione mediante un polinomio di grado inferiore al numero di punti.

Fattorizzazione QR:

Permette di ricondurre il problema alla risoluzione di un sistema lineare quadrato di ordine n o al peggio ad un problema di dimensione minore.

Passaggi:

Fattorizzare $A = QR$

Minimizzare la norma 2 è come minimizzarne il quadrato:

$$\|Ax - b\|_2^2 = \|QRx - b\|_2^2 = \|Q(Rx - Q^t b)\|_2^2 = \|Rx - Q^t b\|_2^2$$

Ma $R = \begin{pmatrix} \hat{R} \\ 0^t \end{pmatrix} \in M_n(\mathbb{R})$; $Q^t b = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$ con $w_1 \in \mathbb{R}^n$; $w_2 \in \mathbb{R}^{m-n}$

Riscriviamo la condizione:

$$\|Ax - b\|_2^2 = \left\|\hat{R}x - w_1\right\|_2^2 + \|w_2\|_2^2$$

Quindi il problema si è ridotto a minimizzare $\hat{R}x - w_1$ o, nel caso $\hat{R}$ sia invertibile, alla risoluzione del sistema lineare $\hat{R}x = w_1$

Proposizione;

Esiste ed è unica la soluzione.

Costo computazionale:

$O(mn^2)$

Osservazione:

Questo metodo è meno efficiente di quello che sfrutta la SVD del problema descritta nei paragrafi successivi.

Sistema delle equazioni normali:

Se vogliamo risolvere $Ax = b$ e supponiamo che A sia di rango massimo allora possiamo ricondurci al sistema quadrato:

$A^t Ax = A^t b \leftrightarrow R^t Rx = R^t Q^t x \leftrightarrow R^t \hat{R} = R^t w_1 \leftrightarrow \hat{R} = w_1$

Teorema:

Detto X l'insieme dei vettori complessi che minimizzano $\|Ax - b\|_2$ allora:

$x \in X \leftrightarrow A^H Ax = A^H b$ (Sistema di equazioni normali)

$X \neq \emptyset$

X si riduce ad un solo elemento $\leftrightarrow$ A ha rango massimo

$\exists \check{x} \in X \mid \|\check{x}\|_2 = \min_{x \in X} \|x\|_2$ detta **Soluzione di minima norma**.

Pratica:

Possiamo ricondurci a studiare la matrice definita positiva e quadrata $A^H A =_{caso\ reale} A^t A$

Miglioramento:

Per migliorare i metodi di ricerca sfruttiamo il teorema di decomposizione in valori singolari.

Teorema di decomposizione in valori singolari (SVD) :

Siano $m \geq n$; $A \in M_{m\times n}(\mathbb{R})$, allora $\exists\ U \in M_{m\times m}(\mathbb{R})$ e $V \in M_{n\times n}(\mathbb{R})$ ortogonali e $\sum \in M_{m\times n}(\mathbb{R})$

con $\sum = \begin{pmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \\ & 0 & & \end{pmatrix}$; $\sigma_i \geq \sigma_j \geq 0\ \forall\ i < j$ e vale:

$$A = U\sum V^t$$

Osservazione approssimazione:
Conviene approssimata A con B meglio condizionata perché si perde il quadrato al numero di condizionamento.

Osservazione rango:
Se A ha rango k allora $\sigma_1, \dots, \sigma_k$ sono diversi da 0 mentre $\sigma_{k+1} = \cdots = \sigma_n = 0$

Decomposizione spettrale:

$$A^tA = V^t\sum U^tU\sum V = V^t\begin{pmatrix} \sigma_1^2 & & & \\ & \sigma_2^2 & & \\ & & \ddots & \\ & & & \sigma_n^2 \end{pmatrix}V$$

Quindi dati gli autovalori λ_i, reali poiché A^tA è simmetrica, ordinati in modo decrescente vale:

$$\sigma_i = \sqrt{\lambda_i}$$

Osservazione algoritmo:
Sebbene mal condizionato e con costo $O(mn^2)$ possiamo calcolare QR di A^tA e da li ricaviamo gli autovalori.

Proprietà principali SVD:
Data $A \in M_{m\times n}(\mathbb{R})$ di rango k allora i σ_i sono nulli dal $(k+1)$-esimo in poi.
Inoltre:

$$Ax = U\sum V^tx = \sum_{i=1}^{k} \sigma_i U_i V_i^t x$$

Quindi:

$$\ker(A) = \langle V_{k+1}, \dots, V_n \rangle$$
$$\mathrm{Imm}(A) = \langle U_1, \dots, U_k \rangle$$

Utilizzo della SVD nel problema dei minimi quadrati:

Usiamo questa fattorizzazione nel problema ai minimi quadrati:

$$\|Ax - b\|_2^2 = \|U\sum V^t x - b\|_2^2 = \|\sum V^t x - U^t b\|_2^2$$

Con $\text{rango}(A) = k < n$
Se chiamiamo $z = V^t x$; $U^t b = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$ otteniamo:

$$\left\|\sum z - \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}\right\|_2^2 = \left\|\begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_k \end{pmatrix} z - y_1\right\|_2^2 + \|y_2\|_2^2$$

Osservazione:
In questo passaggio abbiamo sfruttato il fatto che la lunghezza di y_1 sia proprio k e che $\sum z$ abbia gli autovalori nulli dal $(k+1)$-esimo in poi.
Infatti se $\text{rango}(A) = k < n$ allora partizioniamo il problema in uno di grado $n - k$ e annulliamo le componenti rimaste (Potremo sceglierle liberamente ma così teniamo bassa la norma).

Soluzione esplicita:
$z_j = \frac{y_{1,j}}{\sigma_j} = \frac{U_j^t b}{\sigma_j}$ con U_j j-esima colonna di U.

Se A è di rango massimo:
$x = Vz = \left(\sum_{k=1}^{n} \frac{V_k U_k^t}{\sigma_k}\right) b = A^+ b$

Pseudo inversa di Moore – Penrose:

$A^+ = \left(\sum_{k=1}^{n} \frac{V_k U_k^t}{\sigma_k}\right)$ con n rango.

Equivalente:

$$A^+ = V \sum^+ U^t \text{ con } \sum^+ = \begin{pmatrix} \frac{1}{\sigma_1} & & \\ & \ddots & \\ & & \frac{1}{\sigma_n} \\ & & 0 \end{pmatrix}$$

Perciò x che minimizza la norma 2 e risolve il sistema lineare approssimato è dato proprio da $x = A^+ b$

Osservazione:

$A^+ A = \mathrm{Id}$

Se il rango della matrice A è minore possiamo definire comunque:

$A^+ = \left(\sum_{k=1}^{\mathrm{rango}(A)} \frac{V_k U_k^t}{\sigma_k}\right)$

Condizionamento mediante pseudo inversa:

Per il numero di condizionamento vale:

$K(A) = \|A^+\|_2 \cdot \|A\|_2 = \frac{\sigma_1}{\sigma_k}$ (k rango di A)

Osservazione:

$K_2(A^T A) = K_2(A)^2$

Metodi di calcolo:

Osserviamo che i σ_i sono le radici degli autovalori di A^tA (Semidefinita positiva).

Passi:

Calcoliamo A^tA
Determiniamo la decomposizione spettrale $A^tA = VDV^t$
Calcoliamo la fattorizzazione QR di $AV = UR$

Problema:
Questo metodo prevede il calcolo di A^tA che può essere oneroso.

Osservazione:
Possiamo individuare due matrici ortogonali U, V e una B bidiagonale tali che:
$A = UBV^t$

Metodo per determinarle (mediante matrici di Householder):
Calcoliamo P_1 | $P_1Ae_1 = ae_1$
Calcoliamo Q_1 | P_1AQ_1 abbia la prima riga nulla ad esclusione dei primi due elementi.
Iteriamo sulle righe e colonne successive.

Conclusione:
Vale allora $A^tA = V^tB^tBV$ quindi possiamo cercare la decomposizione spettrale di B^tB (Tridiagonale simmetrica) che è più semplice da determinare.

Costo computazionale:
Calcolare due matrici di Householder $O(n)$
Ripeterlo per ogni passo, dunque $O(n^2)$
Effettuare le moltiplicazioni $O(mn^2)$

Calcolare gli autovalori mediante il metodo QR costa $O(n^2)$ lasciando quindi invariato il costo totale.
Sembrerebbe di dover calcolare B^tB ma in realtà esiste un metodo che permette di risparmiare questo calcolo.
Costo computazionale totale $O(mn^2)$

Stima dell'errore:
Il teorema di Bauer – Fike è applicabile e fornisce una maggiorazione dell'errore assoluto, non abbiamo però maggiorazioni dell'errore relativo che può essere un problema per valori piccoli.

TSVD e applicazioni:

Idea:
La T indica il troncamento del SVD.
Ricordando che i σ_i sono disposti in ordine quindi un modo per abbassare il condizionamento del sistema $Ax = b$ può essere trascurare gli autovalori più bassi azzerandoli.

Risolviamo dunque un sistema troncato simile al precedente:

$$\hat{A}x = U\hat{\sum}V^t = b$$

Attenzione:
Stiamo approssimando l'operatore definito della matrice, non il problema.

Osservazione (Soglia o filtro):
Possiamo azzerare tutti quegli autovalori che sono al di sotto di un valore fissato s così da ottenere una maggiorazione del numero di condizionamento pari a $\sigma_1 s^{-1}$

Osservazione (Risparmiare memoria):
Possiamo troncare anche per risparmiare memoria:

Metodo:

Calcoliamo la SVD: $A = U\sum V^t$

Tronchiamo la $\sum$ ponendo $\sigma_{k+1} = \cdots = \sigma_n = 0$

Approssimiamo il sistema con $B = U\hat{\sum}V^t = \sum_{i=1}^{k} \sigma_i U_i V_i^t$

Osservazione 1 (Memoria occupata):
Con questo troncamento riduciamo di circa $\frac{k}{n}$ lo spazio occupato.

Osservazione 2 (Differenza fra le matrici A e B):
$\|A - B\|_2 = \left\|U\sum V^t - U\hat{\sum}V^t\right\|_2 = \left\|U\left(\sum - \hat{\sum}\right)V^t\right\|_2 = \sigma_{k+1}$
Questa approssimazione inoltre è la migliore possibile di rango k.

Se quindi i σ_i sono divisi in un gruppo dai valori più alti ed uno di quelli più bassi troncando completamente quelli bassi otteniamo un errore piccolo a fronte di una riduzione considerevole della memoria occupata.

Questo è il caso della compressione di una foto dove eliminiamo i valori più bassi.

Capitolo 10: Matrici strutturate di Hessenberg ed Hermitiane

Idea:
Descriviamo questi due tipi di matrici e identifichiamo degli algoritmi per ricondurci allo studio di quest'ultime.

Osservazioni introduttive sulla riducibilità:
Per una matrice di Hessenberg o in particolare per una tridiagonale vale:
La matrice è riducibile $\rightarrow \exists B_i = 0$ con B_j elementi della sottodiagonale.
Una matrice è **non riducibile** se tutti i B_i sono diversi da 0. (Se simmetrica è equivalente ad irriducibile)
Una matrice è **irriducibile** se il suo grafo è fortemente connesso.

Matrici in forma di Hessenberg superiore:

Sono matrici tali che $a_{ij} = 0$ se $i > j + 1$.

$$H = \begin{pmatrix} * & \dots & \dots & * \\ * & \ddots & & \vdots \\ 0 & \ddots & \ddots & \vdots \\ 0 & 0 & * & * \end{pmatrix}$$

Teorema costruttivo:

Per ogni matrice A esistono vari metodi che la rendano in forma di Hessenberg superiore.
(Il più usato è quello di Householder)

Metodo di Gauss:
Applicare le matrici elementari di Gauss (Con tecnica del massimo pivot) per ricondurci alla forma richiesta.

> **Osservazione:**
> Il costo computazionale è di $\frac{5}{6}n^3$ inferiore a quello dei metodi successivi ma presenta problemi di instabilità, soprattutto nel caso in cui gli autovalori della matrice A_k ottenuta dopo k iterazioni abbia autovalori in modulo molto alti rispetto a quelli di A.

Metodo di Givens:
Sfrutta delle matrici con elementi in forma di seno e coseno. Non approfondito.

> **Osservazione:**
> Costo $\frac{10}{3}n^3$

Metodo di Householder:

Ricordiamo che P si dice matrice elementare di Householder se $\exists\ \sigma \in \mathbb{R}\backslash\{0\}$ e $u \in \mathbb{R}^n \mid P = \mathrm{Id} - \sigma u u^H$ e $\sigma = \frac{2}{\|u\|_2}$.

Proprietà matrici di Householder:

P è Hermitiana

P è unitaria

$\det(P) = -1$

Osservazione riducibilità:

Se la matrice A è riducibile studiamo i problemi più piccoli in essa contenuti, supponiamo dunque di esserci già ridotti al caso in cui la matrice sia irriducibile.

Data dunque la generica matrice $A = \begin{pmatrix} a_{11} & w^H \\ v & \hat{A} \end{pmatrix}$ irriducibile.

Sappiamo che esiste una matrice $\hat{P}$ si Householder tale che $\hat{P}v = \beta e_1$ quindi, detta $P = \begin{pmatrix} 1 & 0^H \\ 0 & \hat{P} \end{pmatrix}$ vale:

$$PAP^H = \begin{pmatrix} a_{11} & w^H\hat{P}^H \\ \hat{P}v & \hat{P}A\hat{P}^H \end{pmatrix}$$

Iterando per $n-2$ volte questo passaggio si ottiene una matrice di Hessenberg superiore.

Osservazione Hermitiane:

Se la matrice di partenza era Hermitiana lo è anche quella di arrivo, dunque la matrice conclusiva sarà tridiagonale.

Valutazione costo computazionale:

Il calcolo di $\hat{P}$ è $O(n)$

Il calcolo di $w^H\hat{P}$ è $O(n)$ (Se A non è Hermitiana)

Il calcolo di $\hat{P}A\hat{P}^H$ è $O(n^2)$

In totale l'algoritmo è $\frac{5}{3}n^3 \sim O(n^3)$

Questo algoritmo è stabile all'indietro.

Test sugli 0:

Siccome a meno di precisione di macchina non siamo certi di individuare dei veri 0 su di un calcolatore allora li consideriamo nulli se:

$|B_k| \leq u(|a_k| + |a_{k+1}|)$

Matrici Hermitiane tridiagonali:

Sono matrici Hermitiane tali che $a_{ij} = 0$ se $|i-j| \geq 2$ e tale che $a_{ij} = \overline{a_{ji}}\ \forall\ i,j \in \{1,2,\dots,n\}$

$$T = \begin{pmatrix} a_1 & \overline{b_1} & 0 & 0 \\ b_1 & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \overline{b_{n-1}} \\ 0 & 0 & b_{n-1} & a_n \end{pmatrix}$$

Hermitiane diagonalizzabili:
Una matrice tridiagonale Hermitiana irriducibile ha tutti gli autovalori di molteplicità 1, dunque è diagonalizzabile.

Teorema costruttivo:
Ogni matrice Hermitiana può essere ricondotta mediante una trasformazione ad una matrice Hermitiana tridiagonale.

Metodi:
Tutti questi metodi sfruttano trasformazioni per similitudine unitarie, dunque per il teorema di Bauer – Fike la ricerca degli autovalori è un problema ben condizionato.

Metodo di Householder:
Il metodo descritto nel paragrafo precedente per determinare la forma di Hessenberg di una matrice qualsiasi se viene applicato ad una matrice Hermitiana restituisce una matrice Hermitiana tridiagonale.

> Osservazione:
> Costo $\frac{2}{3}n^3 \sim O(n^3)$

Metodo di Givens:
Sfrutta delle matrici con elementi in forma di seno e coseno. Non approfondito.

> **Osservazione:**
> Costo $\frac{4}{3}n^3$
> Come stabilità è analogo al metodo di Householder ma risulta migliorabile per sfruttare una struttura con abbastanza zeri della matrice.

Metodo di Lanczos:

Idea:
Metodo alternativo a quello di Householder per tridiagonalizzare una matrice simmetrica.
Metodo non realmente usato per effettuare tridiagonalizzazione in quanto non è numericamente stabile e Q così ottenuta non è ortogonale (Scegliendo $q_1 = e_1$ differisce per una matrice di fase da Q ottenuta con il metodo di Householder).
Il metodo di Lanczos viene usato dopo la definizione dei coefficienti di Rayleigh poiché ci da una rapida stima della matrice per approssimare gli autovalori di modulo minimo (**Metodi di Krylov**)

Data $A \in M_n(\mathbb{C})$ Hermitiana allora cerchiamo $Q = (q_1 \quad \cdots \quad q_n)$ unitaria tale che:

$$Q^H A Q = T = \begin{pmatrix} a_1 & b_1 & 0 & 0 \\ b_1 & a_2 & \ddots & 0 \\ 0 & \ddots & \ddots & b_{n-1} \\ 0 & 0 & b_{n-1} & a_n \end{pmatrix} \text{ con } a_i, b_j \in \mathbb{R}\,;\, b_j \geq 0$$

L'idea è che individuato il primo vettore q_1 possiamo trovare tutti gli altri.

Metodo:
Da $AQ = QT$ segue:

$$\begin{cases} Aq_1 = a_1 q_1 + b_1 q_2 \\ Aq_i = b_{i-1} q_{i-1} + a_i q_i + b_i q_{i+1} \\ Aq_n = b_{n-1} q_{n-1} + a_n q_n \end{cases}$$

Dunque:

$$\begin{cases} a_1 = q_1^H A q_1\,;\; q_2 = \frac{(A - a_1 \mathrm{Id}) q_1}{b_1}\,;\, b_1 = \|(A - a_1 \mathrm{Id}) q_i\|_2 \\ a_i = q_i^H A q_i\,;\; q_2 = \frac{(A - a_i \mathrm{Id}) q_i - b_{i-1} q_{i-1}}{b_i}\,;\, b_1 = \|(A - a_i \mathrm{Id}) q_i - b_{i-1} q_{i-1}\|_2 \\ a_n = q_n^H A q_n \end{cases}$$

Osservazione (Scelta del vettore iniziale):
Si può scegliere un vettore qualsiasi $u \in \mathbb{C}^n \mid \|u\|_2 = 1$, scegliendo se uno dei b_i risultasse nullo, come q_{i+1} un vettore ortonormale a tutti quelli prima calcolati.

Proposizione:
I vettori q_i così calcolati sono ortonormali.

Osservazione (Costo computazionale e instabilità):
Interessante è il fatto che se siamo interessati solo alla matrice T dobbiamo limitarci a tenere in memoria solo due vettori per implementare l'algoritmo.

Se A non è sparsa il costo è $O(n^3)$
Se A è a banda con $2p+1$ diagonali il costo è $(2p+6)n^2 \sim_{p \ll n} O(n^2)$

Il metodo di triagonalizzazione di Lanczos presenta problemi di instabilità numerica nel caso in cui uno dei b_i sia piccolo, inoltre Q così calcolata non abbiamo garanzie che sia ortogonale. È utile per il calcolo degli autovalori estremi dello spettro della matrice.

<u>Quoziente di Rayleigh:</u>

Il quoziente di Rayleigh di A e x è lo scalare:

$$R(A,x) = \frac{x^t A x}{x^t x}$$

Osservazione:
Il minimo quoziente di Rayleigh su tutto $\mathbb{R}^n$ è il modulo dell'autovalore minimo.
Il massimo quoziente di Rayleigh su tutto $\mathbb{R}^n$ è il modulo dell'autovalore massimo.

Dato un generico sottospazio $S \subseteq \mathbb{R}^n$ allora:

$\lambda \coloneqq \min_{x \in S} R(A,x)$ è il minimo modulo degli autovalori di S.

Osservazione:
Se S è grande può essere una stima del modulo dell'autovalore minimo di $\mathbb{R}^n$

Osservazione:
Detta $\{q_1 \dots, q_k\}$ base di S e $Q = (q_1 \quad \cdots \quad q_k)$ allora $\forall\, x \in S$ vale:

$$x = \sum_{i=1}^{k} a_i q_i$$

Detto $a = (a_1 \quad \cdots \quad a_k)$ vale $x = Qa$.
Quindi:

$$\frac{x^t A x}{x^t x} = \frac{a^t Q^t A Q a}{a^t Q^t Q a} = \frac{a^t Q^t A Q a}{a^t a}$$

Perciò ci basta minimizzare $R(Q^t A Q, a)$ su $\mathbb{R}^k$ e quindi lavoriamo con matrici $k \times k$ riducendo la complessità del problema.

Definizione (Sottospazio di Krylov):

Sia $A \in M_n(\mathbb{R})$ (Alla stessa maniera è possibile definirlo a partire da $A \in M_n(\mathbb{C})$) e $v \in \mathbb{R}^n$.

Il sottospazio di Krylov di A e v di ordine j è:

$$K_j(A, v) \coloneqq \langle v, Av, \dots, A^{j-1}v \rangle$$

Osservazione:

$\dim\left(K_j(A, v)\right) \leq j$

Metodo:

Dato uno spazio di Krylov scriviamo la fattorizzazione QR della matrice dei vettori che lo generano:

$$(v \quad Av \quad \dots \quad A^{j-1}v) = QR$$

Proposizione:

Le prime j colonne di Q generano di $K_j(A, v)$

Osservazione:

Calcolando $Q^t A Q = B = \left(b_{i,j}\right)$ si ottiene la matrice $k \times k$ simmetrica.

Se $i < (j-1) \rightarrow b_{i,j} = q_i^t A q_j \in K_j(A, v)$ e siccome Aq_j è ortogonale a $q_i \rightarrow b_{i,j} = 0 \rightarrow B$ è tridiagonale.

Collegamento con il metodo di Lanczos:

B si può ottenere con $k-1$ iterazioni del metodo di Lanczos e ci permette di dare una stima degli autovalori di modulo minimo e massimo.

Capitolo 11: Polinomio caratteristico e autovalori

Idea:
Introduciamo lo studio degli autovalori ed autovettori di matrici con alcuni teoremi che offrono garanzie sull'errore totale per lo studio di questi problemi.
Osserviamo anche che lo studio verrà svolto prevalentemente su matrici di Hessenberg e su tridiagonali Hermitiane (Previa tridiagonalizzazione su Hermitiane qualsiasi).

Teorema di Bauer – Fike:
Se $A \in M_n(\mathbb{R})$ (Va bene anche $M_n(\mathbb{C})$) è una matrice diagonalizzabile tramite il cambio di base in V^{-1} (Ossia $A = VDV^{-1}$), $\| \ \|$ una norma assoluta e η è un autovalore della matrice perturbata $A + \delta A$, allora esiste un autovalore λ di A tale che:

$$|\lambda - \eta| \leq \|\delta A\| \cdot K(V)$$

Definizione (Norma assoluta):
Una norma si dice assoluta se $\forall\ D = (d_i) \in M_n(\mathbb{C})$ diagonale vale $\|D\| = \max|d_i|$

Osservazione:
$\| \ \|_1$; $\| \ \|_2$; $\| \ \|_\infty$ sono norme assolute.

Ottimizzare la maggiorazione:
Ricordando che $K(V) = \|V\| \cdot \|V^{-1}\|$ è il numero di condizionamento della matrice per migliorare l'approssimazione dobbiamo ridurre $K(V)$.

Per ogni V vale $K(V) \geq 1$
Se stiamo applicando più cambi di basi $K(VS) < K(V)K(S)$
Se V è una matrice unitaria $K(V) = 1$

Attenzione:
Stiamo sempre parlando di norma Euclidea quando affermiamo le matrici unitarie hanno condizionamento 1.

Quindi dobbiamo cercare di effettuare dei cambi di base unitari.

Osservazione:
$A \in M_n(\mathbb{C})$ è diagonalizzabile mediante matrici unitarie $\leftrightarrow$ A è normale $(AA^H = A^HA)$
Dunque in questi casi $|\lambda - \eta| \leq \|\delta A\|$ ed il problema è ben condizionato.

Errore assoluto e relativo:
La maggiorazione è in valore assoluto e non ci da informazioni sull'errore relativo. Se l'autovalore è piccolo in modulo l'approssimazione può non essere buona.

Idea (Condizionamento):
Per matrici generiche non è garantito il buon condizionamento del problema, quando possibile dunque non calcoleremo direttamente gli autovalori di una matrice ma ci ricondurremo prima a matrici particolari a noi note.

Osservazione (Matrici non normali):
Se la matrice non è normale il calcolo degli autovalori può non essere ben condizionato, studiamo i casi distinguendo a seconda della molteplicità algebrica.

Teorema (Molteplicità 1):
Sia $A \in M_n(\mathbb{C})$; λ autovalore di A di molteplicità algebrica uno, $x, y \in \mathbb{C}^n$; $\|x\|_2 = \|y\|_2 = 1$ tali che:

$Ax = \lambda x$

$y^H A = \lambda y^H$

Allora $y^H x \neq 0$ ed inoltre esiste nel piano complesso un intorno V dello 0 e una funzione analitica

$\lambda(\varepsilon): V \to \mathbb{C}$:

$\lambda(\varepsilon)$ è autovalore di molteplicità algebrica 1 di $A + \varepsilon F$; $F \in M_n(\mathbb{C})$

$\lambda(0) = \lambda$

$\lambda'(0) = \frac{y^H F x}{y^H x}$

A meno di termini di ordine superiore in ε vale $\lambda(\varepsilon) - \lambda = \varepsilon \cdot \frac{y^H F x}{y^H x}$

Formulazione equivalente:
Data una matrice $A \in M_n(\mathbb{C})$, una matrice di perturbazione εF, un autovalore λ di A e un autovalore η della matrice $A + \varepsilon F$ si ha che, dati y, w autovettori destro e sinistro di modulo 1 relativi a λ:

$$|\lambda - \eta| \leq \frac{1}{w^H y} \cdot \|\varepsilon F\|$$

Conseguenza pratica:
La variazione dell'autovalore dato da una perturbazione della matrice risulta proporzionale al coefficiente di perturbazione ed è tanto maggiore quanto più piccolo è $|y^H x|$.

Osservazione (Molteplicità maggiore):
In questo caso maggiore è la dimensione dei blocchi di Jordan (Ossia maggiore è la disparità fra le molteplicità geometrica e algebrica) più il problema è mal condizionato.

Caso di matrici Hermitiane tridiagonali:

Teorema di Courant - Fisher o del minimax:

Sia $A \in M_n(\mathbb{C})$ una matrice Hermitiana con autovalori:

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$$

Allora risulta:

$$\lambda_{n-k+1} = \min_{V_k} \max_{\substack{x \neq 0 \\ x \in V_k}} r_A(x)$$

$$\lambda_k = \max_{V_k} \min_{\substack{x \neq 0 \\ x \in V_k}} r_A(x)$$

Dove V_k è un qualunque sottospazio di $\mathbb{C}^n$ di dimensione k per $k = 1, \dots, n$

Teorema della perturbazione di rango 1:

Sia $u \in \mathbb{C}^n$, $\sigma \in \mathbb{R}$, $\sigma > 0$, siano $A, B \in M_n(\mathbb{C})$ Hermitiane, tali che:

$$B = A + \sigma \cdot uu^H$$

Per gli autovalori:

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$$

di A e:

$$\mu_1 \geq \mu_2 \geq \cdots \geq \mu_n$$

di B, vale la relazione:

$$\lambda_1 + \sigma \cdot u^H u \geq \mu_1 \geq \lambda_1 \geq \mu_2 \geq \lambda_2 \geq \cdots \geq \mu_n \geq \lambda_n$$

Calcolo del polinomio caratteristico:

Nel caso di matrici Hermitiane tridiagonali la loro struttura particolare suggerisce un metodo rapido per calcolare il polinomio caratteristico (Sviluppando l'ultima riga) e di ogni polinomio caratteristico delle sottomatrici di testa di dimensione k.

$$p_k(\lambda) = (a_k - \lambda) \cdot p_{k-1}(\lambda) - |b_{k-1}|^2 \cdot p_{k-2}(\lambda)$$

Osservazione:

Conviene porre $p_0(\lambda) = 1$; $p_{-1}(\lambda) = 0$ per poterlo applicare al primo passo.

Costo dell'algoritmo $5n \sim O(n)$.

Osservazione interessante:

Se λ è un autovalore per A_k non è un autovalore di A_{k-1}.

Teorema di separazione delle sottomatrici di testa:

Sia $A \in M_n(\mathbb{C})$ Hermitiana e A_k la matrice di testa k-esima.

Allora gli autovalori di A_k separano quelli di A_{k+1}

Equazione secolare (Matrici Hermitiane tridiagonali):

Procedimento (Matrici reali per semplicità di notazione):

1.

Scriviamo la matrice A come:

$$A = \begin{pmatrix} \begin{pmatrix} & A_{n-1} & \end{pmatrix} & \begin{matrix} b_{n-1} \\ a_n \end{matrix} \\ \begin{matrix} & b_{n-1} \end{matrix} & \end{pmatrix}$$ (Arrow matrix)

2.

Sappiamo che esiste una matrice ortogonale Q_{n-1} tale che:

$$\begin{pmatrix} \begin{pmatrix} & Q_{n-1} & \end{pmatrix} & \\ & 1 \end{pmatrix} \begin{pmatrix} \begin{pmatrix} & A_{n-1} & \end{pmatrix} & \begin{matrix} b_{n-1} \\ a_n \end{matrix} \\ \begin{matrix} & b_{n-1} \end{matrix} & \end{pmatrix} \begin{pmatrix} \begin{pmatrix} & Q_{n-1}^T & \end{pmatrix} & \\ & 1 \end{pmatrix} = \begin{pmatrix} \begin{pmatrix} & D_{n-1} & \end{pmatrix} & \begin{matrix} w \\ a_n \end{matrix} \\ \begin{matrix} & w^T \end{matrix} & \end{pmatrix} \coloneqq B_n$$

Con $D_{n-1} = \begin{pmatrix} \lambda_1^{\langle n-1 \rangle} & & \\ & \ddots & \\ & & \lambda_{n-1}^{\langle n-1 \rangle} \end{pmatrix}$

3.

Sfruttando il complemento di Schur sappiamo che esistono delle matrici a blocchi tali che:

$\begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} \mathrm{Id} & 0 \\ * & \mathrm{Id} \end{pmatrix} \begin{pmatrix} A & 0 \\ 0 & T \end{pmatrix} \begin{pmatrix} \mathrm{Id} & * \\ 0 & \mathrm{Id} \end{pmatrix}$ con $T = D - CA^{-1}B$ il Complemento di Schur della matrice.

Questo ci permette di calcolare più facilmente il determinante di B_n ottenuto come:

$p_n(z) = \det(D_{n-1}) \cdot \det(\mathrm{Schur}) = \det(D_{n-1} - z \cdot \mathrm{Id}) \cdot (a_n - z - w^T (D_{n-1} - z \cdot \mathrm{Id}) w)$

Quindi:

$$p_n(z) = \left(\prod_{j=1}^{n-1} \left(\lambda_j^{\langle n-1 \rangle} - z\right)\right) \left(a_n - z - \sum_{i=1}^{n-1} \frac{w_i^2}{\lambda_i^{\langle n-1 \rangle} - z}\right)$$

Conclusione:

Siccome gli autovalori di A_n sono diversi da quelli di A_{n-1} allora le radici del polinomio caratteristico sono le soluzioni dell'equazione secolare:

$$g(z) = a_n - z - \sum_{i=1}^{n-1} \frac{w_i^2}{\lambda_i^{\langle n-1 \rangle} - z} = 0$$

Quindi nel tracciare il grafico (Se gli autovalori sono ordinati decrescenti) basta ricordarsi che:

Ad ogni autovalore di A_{n-1} corrisponde un asintoto verticale.

La derivata di $g(z)$ è sempre negativa e quindi la funzione è decrescente.

Ogni due autovalori di A_{n-1} c'è dunque esattamente un autovalore di A_n.(Proprietà di interlacing)

Osservazione:

Per ora non abbiamo gli strumenti per definire un'iterazione funzionale per il calcolo degli autovalori di A_n a parte il normale metodo di bisezione (Inefficiente) nel caso in cui si conoscano gli autovalori di A_{n-1}. Il metodo di bisezione di Sturm invece è un sistema efficiente.

In generale sulle matrici di Hessenberg non esiste un metodo altrettanto efficace.

Capitolo 12: Metodi numerici per il calcolo di autovalori e autovettori

Idea:
Mostriamo gli algoritmi effettivamente utilizzati nel calcolo degli autovalori e autovettori di matrici Hermitiane e in forma di Hessenberg.

Calcolo degli autovalori:

Metodo di Sturm:
Si applica a matrici Hermitiane tridiagonali (Previa riduzione Hermitiane qualsiasi).
Questo metodo permette di ricavare in maniera esplicita l'i-esimo autovalore (Costo $O(n^2)$ se tridiagonale oppure $O(n^3)$ se da ridurre) o l'intero spettro (Costo $O(n^3)$ in entrambi i casi)

Metodo QR:
Questo è uno dei metodi più efficaci per il calcolo degli autovalori di una matrice qualsiasi mediante l'uso della fattorizzazione QR tramite matrici di Householder.

Vantaggi:
Fornisce l'intero spettro della matrice.
Questo metodo inoltre è applicabile a matrici qualsiasi.
Svantaggi:
Ha delle condizioni di convergenza molto restrittive.

Divide et Impera (Cuppen):
Si applica a matrici Hermitiane tridiagonali (Previa riduzione Hermitiane qualsiasi).
Questo metodo permette di ricavare l'intero spettro passando per il calcolo degli autovettori.

Vantaggio: Otteniamo una base di autovettori e l'intero spettro.
Svantaggio: Il calcolo degli autovettori è un problema mal condizionato e questo condiziona il risultato ottenuto anche sugli autovalori.

Calcolo degli autovettori:

Metodo delle potenze:
Permette di calcolare l'autovettore relativo all'autovalore di modulo massimo o minimo.
Se gli autovalori sono già noti permette di ricavare una base di autovettori.

Metodo di Sturm:

Data una matrice T_n tridiagonale Hermitiana (Mediante riduzione un'Hermitiana qualsiasi).

Osservazioni preliminari:
Per quanto visto prima il costo per valutare il polinomio caratteristico è $O(n)$.
La valutazione di tutti i polinomi p_k in punto x fissato ha costo $O(n)$.
Se la matrice con cui stiamo lavorando è riducibile ci riconduciamo ai sottocasi più piccoli.

Definiamo f:

$f: \mathbb{R} \to \{0,1, \dots, n\}$; $f(x) =$ # cambi di segno nel vettore $\begin{pmatrix} p_0(x) \\ \vdots \\ p_n(x) \end{pmatrix}$

Buona definizione:
Imponiamo che se $p_j(x) = 0$ allora il segno lo consideriamo uguale a $p_{j-1}(x)$.
Osserviamo a questo proposito che $p_0(x) = 1$.

Se f cambia valore in un punto significa che questo deve essere lo 0 di almeno uno dei p_k

Osservazione:
Polinomio di bordo:
$p_0(x) = 1 \ \forall x$
Polinomio interno:
Se $p_j(\bar{x}) = 0$ allora:
$p_{j-1}(\bar{x}), p_{j+1}(\bar{x}) \neq 0$
Da $p_{j+1}(x) = (a_{j+1} - x) \cdot p_j(x) - |b_j|^2 \cdot p_{j-1}(x)$ si ottiene che $p_{j-1}(\bar{x}) \cdot p_{j+1}(\bar{x}) < 0$
Quindi f non cambia di segno se un polinomio interno si azzera in un punto e più in generale non dipende dal segno dei polinomi interni.

Osserviamo che la derivata di p_n e quella di p_{n-1} sono opposte in segno, essendo gli 0 semplici allora la derivata non è nulla.

Quindi ad ogni cambio di p_n cambia il valore di f.
In conclusione f rappresenta il numero di 0 di p_n, ossia del polinomio caratteristico. (Teorema di Sturm)

Polinomio esterno (Polinomio caratteristico):
$\lim_{x \to -\infty} p_n(x) = +\infty$; gli 0 di p_{n-1} separano gli 0 di p_n ; gli 0 di una matrice tridiagonale Hermitiana sono semplici.

Teorema della successione di Sturm:
$f(b) - f(a) =$ # di autovalori, ossia di 0 del polinomio caratteristico in $[a, b[$

Metodo di bisezione di Sturm:

Ipotizziamo di voler calcolare il j-esimo autovalore.

Ricordiamo la disuguaglianza di Hirsch $|\lambda| \leq \|T_n\|$, questo vale per ogni autovalore e quindi anche per il j-esimo.

Calcoliamo $f(0)$ il cui valore indica il numero di autovalori prima di 0.
Se $i \leq f(0)$ allora significa che il nostro autovalore è nell'intervallo $\left[-\|T_n\|, 0\right[$.

Calcoliamo adesso $f\left(-\frac{\|T_n\|}{2}\right)$ e iteriamo il procedimento.

Osservazione (Intervallo e metodo di Newton):
L'intervallo che contiene l'autovalore al passo k-esimo ha dimensione $2^{-k}(b_0 - a_0)$.
Il metodo di Newton è comunque più efficiente nel ricavare con esattezza l'autovalore.

Valutazione costo:
Il costo per ridurre un'Hermitiana in forma tridiagonale è $O(n^3)$.
Il calcolo di un autovalore specifico ha costo $O(n^2)$.
Il calcolo dell'intero spettro (Sia nel caso tridiagonale che in quello generale) ha costo $O(n^3)$.
Questo non è molto efficiente perché si può individuare un algoritmo a costo $O(n^2)$.
Se abbiamo macchine che lavorano in parallelo il metodo di Sturm diventa interessante anche per il calcolo dell'intero spettro (Facendo calcolare a ciascuna macchina una parte degli autovalori).

Teorema di Hirsch:
$\forall\, A \in M_n(\mathbb{C})$, $\|\ \|$ norma qualsiasi, allora l'insieme $\{z \in \mathbb{C} \mid |z| \leq \|A\|\}$ contiene tutti gli autovalori.
In particolare per ogni autovalore vale $|\lambda| \leq \|A\|$.

Metodo di Newton:

Definita la relazione a tre termini precedenti è possibile determinare un autovalore sfruttando il metodo di Newton:

$$\begin{cases} p_0'(x) = 0 \\ p_1'(x) = -1 \\ p_{j+1}'(x) = \left(a_{j+1} - x\right) \cdot p_j(x) - b_j^2 \cdot p_{j-1}(x) \end{cases}$$

Studiamo quindi il problema $f(x) = x - \frac{p_n(x)}{p_n'(x)}$

Attenzione:
Necessita di un raffinamento mediante bisezione di Sturm dell'intervallo di definizione, infatti una volta che ci siamo ridotti ad un intervallo con un solo autovalore all'interno al funzione è monotona e dunque è possibile ricavare l'autovalore con il metodo di Newton. Il costo è $O(n)$ per ogni passo.

Metodo QR:

Idea:
Questo è uno dei metodi più efficaci per il calcolo degli autovalori di una matrice qualsiasi mediante l'uso della fattorizzazione QR tramite matrici di Householder.
Fornisce l'intero spettro della matrice.
Questo metodo inoltre è applicabile a matrici qualsiasi ma il problema è che ha delle condizioni di convergenza molto restrittive.

Metodo di base:
Data una matrice $A \in M_n(\mathbb{C})$ la successione convergente alla matrice diagonale degli autovalori è:

$$\begin{cases} A_0 = A \\ A_{k+1} = R_k Q_k \end{cases}$$

Con $Q_k R_k$ la fattorizzazione QR di A_k.

Osservazione similitudine:
A_{k+1} è simile ad A_k, infatti:

$$A_{k+1} = R_k Q_k = Q_k^H Q_k R_k Q_k = Q_k^H A_k Q_k$$

Siccome le trasformazioni unitarie preservano il condizionamento allora questa è una buona successione.

Teorema di convergenza (Wilkinson):
Se gli autovalori sono distinti in modulo allora il metodo converge.
In questo caso converge significa che $A_k \to T$ (Triangolare)

Osservazione:
Non si riesce ad indebolire le ipotesi.

Costo computazionale:
Empiricamente pare che $O(n)$ approssimi l'intero spettro.
La fattorizzazione QR di una matrice qualsiasi è $O(n^3)$
Il numero di passi per avere convergenza cresce linearmente con la dimensione.
Il costo totale del metodo è dunque $O(n^4)$

Se lavoriamo con matrici tridiagonali Hermitiane il costo della fattorizzazione QR di ogni passo (Poiché otteniamo una successione di matrici tridiagonali Hermitiane) è $O(n)$
Il costo totale del metodo è dunque $O(n^2)$

Per le matrici di Hessenberg il costo della fattorizzazione QR è $O(n^2)$.
Il costo totale del metodo è dunque $O(n^3)$

Osservazione Hessenberg:
$A_k \to$Triangolare superiore dunque i $b_k \to 0$

Ottimizzare la velocità di convergenza:
L'idea è applicare uno shift alla matrice A_k, considerare quindi la matrice:
$A_k - c \cdot \mathrm{Id} = \hat{Q}\hat{R} \rightarrow A_{k+1} = \hat{R}\hat{Q} + c \cdot \mathrm{Id}$

<u>Caso matrici di Hessenberg con valori complessi:</u>

Data $A_k = \begin{pmatrix} a_{1,1}^{(k)} & * & & * & * \\ b_1^{(k)} & \ddots & & * & * \\ & \ddots & & \ddots & * \\ & & b_{n-1}^{(k)} & & a_{n,n}^{(k)} \end{pmatrix}$

Primo autovalore (Metodo di Wilkinson):
Applichiamo l'algoritmo con shift di $-a_{nn}^{(k)} \cdot \mathrm{Id}$

Quando $\left|b_{n-1}^{(k)}\right| \leq u\left(\left|a_{nn}^{(k)}\right| + \left|a_{n-1,n-1}^{(k)}\right|\right)$ con u la precisione di macchina memorizziamo l'approssimazione dell'autovalore ottenuto.

Autovalore successivo (Metodo di Wilkinson):
Iteriamo con il minore di dimensione $n-1$ in alto a sinistra (Che sarà ancora in forma di Hessenberg superiore).

Bulge Chasing:

Nel caso di matrici a valori reali non possiamo applicare il procedimento precedente. Il passaggio ad un'aritmetica complessa rischia infatti di portare ad errori tali da non ottenere più due autovalori complessi coniugati, il che può essere problematico.

Conviene vedere allora il problema nel seguente modo:

$p_k(x) = x - a_k$ dunque $\hat{A}_k = p(A_k)$, nel caso reale però $p_k(z) = (x - \lambda)(x - \bar{\lambda})$

Problema:
Il polinomio essendo reale è di secondo grado e il calcolo di A_k^2 non è agevole.

L'idea è modificare la fattorizzazione QR mentre calcoliamo $p_k(A_k)$.

Procedimento:
Si calcola solo la prima colonna di $p_k(A_k)$, ossia $p_k(A_k)e_1$

Calcoliamo una matrice di Householder P_0 | $P_0 p_k(A_k)e_1 = ae_1$.

Costruiamo le altre $n - 1$ matrici di Householder tali che fissato $Z = P_0 \dots P_{n-1}$ valga:
$A_{k+1} = Z^H A_k Z$.

Questa iterazione produce una matrice simile a meno di una matrice di fase reale rispetto a quella presentata prima.

Metodo Divide et Impera (Cuppen):

Passo 1:

Ricondurci ad una matrice della forma $D + ww^H$ conoscendo gli autovettori

Data la matrice tridiagonale H possiamo scriverla a blocchi più un correttivo di rango 2:

$$H = \begin{pmatrix} T_1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & T_2 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & b_{\frac{n}{2}} & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Oppure a blocchi con un correttivo di rango 1:

$$H = \begin{pmatrix} \widehat{T_1} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \widehat{T_2} \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & b_{\frac{n}{2}} & b_{\frac{n}{2}} & 0 \\ 0 & b_{\frac{n}{2}} & b_{\frac{n}{2}} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Con $\widehat{T_1} = T_1$ tranne il blocco $\hat{a}_{\frac{n}{2}} = a_{\frac{n}{2}} - b_{\frac{n}{2}}$ e $\widehat{T_2} = T_2$ tranne il blocco $\hat{a}_{\frac{n}{2}+1} = a_{\frac{n}{2}+1} - b_{\frac{n}{2}}$

Ipotesi (Conoscenza degli autovettori):

Supponiamo di saper calcolare:

$$\widehat{T_1} = Q_1 D_1 Q_1^H \; ; \widehat{T_2} = Q_2 D_2 Q_2^H$$

Di conseguenza supponiamo di sapere autovettori e autovalori di $\widehat{T_1}$; $\widehat{T_2}$

Siano $Q = \begin{pmatrix} Q_1 & 0 \\ 0 & Q_2 \end{pmatrix}$ e $D = \begin{pmatrix} D_1 & 0 \\ 0 & D_2 \end{pmatrix}$ allora:

$$B = Q^H H Q = D + b_{\frac{n}{2}} \cdot ww^H$$

Con $w = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix}$ con q_i l'ultima colonna di Q_1^H e la prima di Q_2^H

Passo 2: Calcolare gli autovalori

Cerchiamo gli autovalori di $D + \theta \cdot ww^H$:

$$p(\lambda) = \det(\lambda \cdot \mathrm{Id}) \cdot \det(\mathrm{Id} + \theta(\lambda \cdot \mathrm{Id} - D)^{-1} ww^H)$$

Osservando che $(\mathrm{Id} + \theta(\lambda \cdot \mathrm{Id} - D)^{-1} ww^H)$ è una matrice elementare sviluppiamo:

$$p(\lambda) = \prod_{j=1}^{n} (\hat{d}_j - \lambda) + \theta \sum_{k=1}^{n} w_k^2 \prod_{\substack{s=1 \\ s \neq k}}^{n} (\hat{d}_s - \lambda)$$

Con $\hat{d}_i$ gli autovalori della matrice diagonale e w calcolati a partire dalle matrici Q_i prima descritte.

Caso particolare:

Ipotesi aggiuntive:
$w_j \neq 0 \ \ \forall\, j = 1 \dots n$
$\hat{d}_k \neq \hat{d}_j \ \ \forall\, k \neq j$ (Autovalori distinti)

> **Osservazione:**
> Anche nei casi senza ipotesi aggiuntive ci si può ricondurre a questo.

La relazione diventa:

$$p(\lambda) = 0 \leftrightarrow 1 + \theta \sum_{j=1}^{n} \frac{w_j^2}{\hat{d}_j - \lambda} = 0$$

Conseguenza (Bisezione):
Gli autovalori delle matrici di ordine $\frac{n}{2}$ separano quelli della matrice iniziale (Ovviamente a loro volta possiamo suddividerli in matrici più piccole). Da questa osservazione possiamo applicare il metodo di bisezione per individuare gli autovalori λ.

> **Osservazione autovettori:**
> Per poter utilizzare la relazione dobbiamo conoscere i vettori z_i delle matrici più piccole ottenuti a partire da Q_1, Q_2 matrici che diagonalizzano T_1 e T_2, in altre parole dobbiamo trovare gli autovettori.

Passo 3: Calcolare gli autovettori

Osservazione:

Noti gli autovalori il calcolo di un autovettore ha costo $O(n)$.

Il calcolo degli autovettori è mal condizionato, inoltre sfruttandoli noi per il passaggio successivo (Abbiamo spezzato il problema e dobbiamo "risalire") non possiamo garantire l'accuratezza del risultato finale.

Metodo:

Cerchiamo gli autovettori di $\widehat{D} + b_{\frac{n}{2}} z z^H$ e una volta ottenuta la matrice da questi U moltiplicarla per la matrice Q.

Formula esplicita:

Data $\widehat{D} + \theta \cdot z z^H$

$v_j = \frac{-\theta z_j}{d_j - \lambda}$

Motivazione:

Se v è autovettore per $D + \theta z z^H$ allora:

$$\left(\widehat{D} + \theta z z^H - \lambda \cdot \mathrm{Id}\right) v = 0$$

Quindi:

$$\left(\widehat{D} - \lambda \cdot \mathrm{Id}\right) v = -\theta z z^H v$$

Siccome $z^H v \neq 0$ possiamo assumere $z^H v = 1$, si ottiene allora:

$$v = -\theta z \left(\widehat{D} - \lambda \cdot \mathrm{Id}\right)^{-1}$$

Equivalente:

$$v_j = \frac{-\theta z_j}{d_j - \lambda}$$

Problemi del metodo:

Usando gli autovettori per procedere nel metodo non abbiamo garanzie sull'accuratezza del risultato.

La matrice U deve essere ortogonale ma l'accumularsi dell'errore può inficiare questa proprietà.

Il calcolo di $\widehat{Q}U$ in generale ha costo non trascurabile.

Siccome U è una matrice Cauchy – Like il costo del prodotto è $O(n^2 \log n)$

L'algoritmo si applica solo ad Hermitiane tridiagonali (O aumentandone il costo ad Hermitiane qualsiasi), non esiste un'implementazione robusta per le matrici di Hessenberg.

Metodo delle potenze:

Idea:
Questo metodo permette di calcolare velocemente l'autovettore relativo all'autovalore di modulo massimo sotto le ipotesi aggiuntive che sia semplice e che sia strettamente maggiore in modulo degli altri autovalori.
Con alcune modifiche si può calcolare ogni altro autovettore.
Necessita di una stima degli autovalori.

Teorema di convergenza:
Se la matrice è diagonalizzabile e ha un autovalore in modulo strettamente maggiore di ogni altro allora il metodo converge.

Metodo diretto:
Scriviamo la successione:

$$\begin{cases} y_0 = \text{liberamente NON autovettore} \\ y_k = Ay_{k-1} \end{cases}$$

> **Attenzione:**
> Se y_0 anche fosse un autovettore dopo qualche iterazione non lo sarebbe più.

Scrivendolo nella base di autovettori $y_0 = \sum a_i v_i$ dunque:

$$y_k = \lambda_1^k \left(a_1 v_1 + \sum_{i=2}^{n} \left(\frac{\lambda_i}{\lambda_1}\right)^k a_i v_i \right) \to \lambda_1^k a_1 v_1$$

Osservazione:
Per stimare l'autovalore basta studiare:

$$\lim_{k\to+\infty} \frac{y_i^{(k+1)}}{y_i^{(k)}} = \lambda_1$$

Per l'autovettore invece osserviamo che:

$$\lim_{k\to+\infty} \frac{y_k}{\lambda_1^k} = a_1 x_1$$

> **Equivalente:**
> $$\lim_{k\to+\infty} \frac{y_j^{(k)}}{\lambda_1^k} = a_1 x_j^{(1)}$$
> $$\lim_{k\to+\infty} \frac{y_k}{y_j^{(k)}} = \frac{x_1}{x_j^{(1)}}$$

Problemi di Overflow:
Per ovviare al problema di Overflow possiamo normalizzare ad ogni passo il vettore.

Metodo diretto:

$$\begin{cases} u_k = At_{k-1} \\ t_k = \frac{1}{b_k} u_k \end{cases}$$

Con b_k normalizzatore, la successione t_k converge all'autovettore x_1 normalizzato.

Metodo inverso (Variante di Wielandt):
Se vogliamo calcolare l'autovettore relativo all'autovalore di modulo minore possiamo studiare:

$$\begin{cases} y_0 \\ y_k = b_k A^{-1} y_{k-1} \end{cases}$$

In forma di sistema lineare:

$$\begin{cases} y_0 \\ Az_k = y_{k-1} \\ y_k = \frac{1}{b_k} z_k \end{cases}$$

Con b_k scalare che normalizzi y_k.

Risolvibile con metodi diretti o iterativi.

Valutazione costo calcolo autovalore:
Per matrici Hermitiane tridiagonali è $O(n)$
Per le matrici di Hessenberg è $O(n^2)$

Metodo inverso con shift:
Se abbiamo una buona stima di un autovalore λ_i possiamo applicare il metodo delle potenze inverse alla matrice $A - \lambda_i \cdot \mathrm{Id}$

$$\begin{cases} y_0 \\ (A - \lambda_i \cdot \mathrm{Id}) z_k = y_{k-1} \\ y_k = \frac{1}{b_k} z_k \end{cases}$$

Osservazione:
Dal punto di vista computazionale conviene fare la fattorizzazione LU delle matrici e poi iterare sfruttando quelle.

Valutazione costo computazionale tutti gli autovettori:
Per matrici Hermitiane tridiagonali è $O(n^2)$
Per le matrici di Hessenberg è $O(n^3)$

Matrici di Toeplitz:

Definizione (Matrice di Toeplitz):
$A \in M_n$ si dice di Toeplitz se le sue diagonali sono costanti ossia$\forall\, i,j$, $\forall\, k \in \mathbb{Z}$ vale $a_{i,j} = a_{i+k,j+k}$

$$A = \begin{pmatrix} a & d & e \\ b & a & d \\ c & b & a \end{pmatrix}$$

Proprietà:
Sono matrici per cui è facile calcolare il prodotto matrice vettore. Ad esempio nel caso di matrici di Toeplitz triangolari inferiori.

$$\begin{pmatrix} t_0 & & & \\ t_1 & \ddots & & \\ \vdots & \ddots & \ddots & \\ t_n & \dots & t_1 & t_0 \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ \vdots \\ p_n \end{pmatrix} = \begin{pmatrix} t_0 p_0 \\ t_1 p_0 + t_0 p_1 \\ \vdots \\ t_n p_0 + t_{n-1} p_1 + \cdots + t_0 p_n \end{pmatrix}$$

Che è come calcolare il prodotto fra i due polinomi, quindi possiamo usare la trasformata discreta di Fourier.

Costo computazionale:
Prodotto vettore matrice mediante trasformata discreta di Fourier costa $O(n \log n)$
Metodo del gradiente coniugato costa $O(n^2 \log n)$

Caso particolare: Matrici di Toeplitz tridiagonali simmetriche

Idea:
Questo caso particolare in realtà si presenta quando discretizziamo il problema differenziale $\Delta u = f$.

Sia $T = \begin{pmatrix} a & b & & \\ b & \ddots & \ddots & \\ & \ddots & \ddots & b \\ & & b & a \end{pmatrix}$ una matrice di Toeplitz tridiagonale simmetrica.

Studiamone il condizionamento (In norma 2) che influenza la convergenza del metodo del gradiente coniugato per il teorema dell'errore.

$$(T - \lambda \cdot \mathrm{Id})x = 0 \leftrightarrow \begin{pmatrix} a & b & & \\ b & \ddots & \ddots & \\ & \ddots & \ddots & b \\ & & b & a \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \bar{0}$$

Scrivendo la relazione a tre termini otteniamo la relazione:

$$1 + \frac{a-\lambda}{b} + x^2 = 0$$

Usando ad esempio i teoremi di Gershgorin possiamo fornire maggiorazioni al condizionamento della matrice.