# LEZIONI DI CALCOLO NUMERICO

Luca Gemignani
Dipartimento di Informatica
Università di Pisa
email : luca.gemignani@unipi.it
url : http://pages.di.unipi.it/gemignani/

# Indice

**1 Introduzione** **2**

**2 L'Aritmetica del Calcolatore** **3**
Lezione 2.1: Rappresentazione in Base e Numeri di Macchina. . . . . 3
Lezione 2.2: Aritmetica di Macchina. . . . . 5
Lezione 2.3: Esercizi. . . . . 7

**3 Analisi degli Errori** **9**
Lezione 3.1: Errori nel Calcolo di una Funzione Razionale. . . . . 9
Lezione 3.2: Tecniche per l'Analisi degli Errori. . . . . 11
Lezione 3.3: Cenni sul Calcolo di una Funzione non Razionale. . . . . 13
Lezione 3.4: Esercizi. . . . . 14

**4 I Problemi dell'Algebra Lineare Numerica: Aspetti Computazionali e Condizionamento** **17**
Lezione 4.1: Norme Matriciali e Norme Vettoriali. . . . . 17
Lezione 4.2: Il Problema della Risoluzione di un Sistema Lineare ed il suo Condizionamento. . . . . 20
Lezione 4.3: Il Problema del Calcolo degli Autovalori di una Matrice ed il suo Condizionamento. . . . . 22
Lezione 4.4: Teoremi di Localizzazione per Autovalori. . . . . 24
Lezione 4.5: Esercizi. . . . . 26

**5 Metodi Diretti per la Risoluzione di Sistemi Lineari** **29**
Lezione 5.1: Sistemi Triangolari. . . . . 29
Lezione 5.2: Matrici Elementari di Gauss ed il Metodo di Eliminazione Gaussiana. . . . . 31
Lezione 5.3: Il Metodo di Gauss per Matrici Invertibili: Tecniche di Pivoting e Stabilità. . . . . 34
Lezione 5.4: Esercizi. . . . . 37

**6 Metodi Iterativi per la Risoluzione di Sistemi Lineari** **44**
Lezione 6.1: Generalità sui Metodi Iterativi. . . . . 44
Lezione 6.2: I Metodi di Jacobi e Gauss-Seidel. . . . . 46
Lezione 6.3: Convergenza dei Metodi di Jacobi e Gauss-Seidel. . . . . 49
Lezione 6.4: Raffinamento Iterativo. . . . . 50
Lezione 6.5: Esercizi. . . . . 51

**7 Calcolo di Autovalori ed Autovettori: Il Metodo delle Potenze 57**
Lezione 7.1: Generalità sul Metodo delle Potenze. . . . . 57
Lezione 7.2: Approssimazione dell'Autovalore Dominante. . . . . 59
Lezione 7.3: Approssimazione dell'Autovettore. . . . . 61
Lezione 7.4: Varianti del Metodo delle Potenze. . . . . 62
Lezione 7.5: Esercizi. . . . . 63

**8 L'algoritmo di PageRanking 67**

**9 Alcuni Problemi Numerici in Teoria dell'Approssimazione 70**
Lezione 9.1: Introduzione. . . . . 70
Lezione 9.2: Il Problema del Calcolo degli Zeri di una Funzione. . . . . 71
Lezione 9.3: Il Problema dell'Approssimazione Polinomiale di una Funzione. . . . . 72
Lezione 9.4: Il Problema del Calcolo dell'Integrale Definito di una Funzione. . . . . 73

**10 Metodi Numerici per l'Approssimazione degli Zeri di una Funzione 74**
Lezione 10.1: Il Metodo di Bisezione. . . . . 74
Lezione 10.2: Metodi di Iterazione Funzionale. . . . . 76
Lezione 10.3: Il Metodo delle Tangenti. . . . . 78
Lezione 10.4: Il Caso delle Equazioni Algebriche. . . . . 80
Lezione 10.5: Esercizi. . . . . 81

**11 Interpolazione Polinomiale ed Integrazione Numerica 91**
Lezione 11.1: Il Problema dell'Interpolazione Polinomiale. . . . . 91
Lezione 11.2: Resto dell' Interpolazione Polinomiale. . . . . 93
Lezione 11.3: Integrazione Numerica. . . . . 94
Lezione 11.4: Esercizi. . . . . 95

**12 Politiche di Vaccinazione e Modelli Epidemiologici 104**

# Capitolo 1

# Introduzione

Queste note raccolgono le lezioni di calcolo numerico tenute dal docente a Pisa nei corsi di studio di laurea triennale di informatica e di ingegneria biomedica. L'insegnamento intende fornire un'introduzione al calcolo scientifico. Vengono illustrati metodi numerici di base per la risoluzione approssimata con il calcolatore di problemi matematici di interesse applicativo che per dimensioni o complessità non possono essere risolti con carta e penna. Tra questi citiamo il calcolo degli zeri di funzioni, la risoluzione di sistemi di equazioni lineari, il calcolo degli autovalori di matrici, l'integrazione definita e l'approssimazione di funzioni e di dati. La trattazione di ogni argomento ricerca un equilibrio tra il rigore matematico e le approssimazioni introdotte di volta in volta per definire metodi ed algoritmi di risoluzione numerica. L'enfasi è posta sull'analisi degli aspetti computazionali, quali il condizionamento dei problemi esaminati e la stabilità e la complessità dei metodi proposti, che assumono rilevanza una volta si proceda al calcolo approssimato mediante calcolatore. Le esercitazioni con l'ausilio dello strumento di calcolo MATLAB introducono lo studente all'analisi sperimentale degli algoritmi e alla validazione dei risultati.

# Capitolo 2

# L'Aritmetica del Calcolatore

## Lezione 2.1: Rappresentazione in Base e Numeri di Macchina.

Sia $B \in \mathbb{N}$, $B > 1$. Il seguente teorema caratterizza la rappresentazione di un numero $x \in \mathbb{R}$, $x \neq 0$, in base $B$.

**Teorema 2.1.1.** Dato $x \in \mathbb{R}$, $x \neq 0$, esistono e sono univocamente determinati

1. un intero $p \in \mathbb{Z}$ detto *esponente* della rappresentazione;
2. una successione di numeri naturali $\{d_i\}_{i \geq 1}$, con $d_1 \neq 0$, $0 \leq d_i \leq B-1$ e $d_i$ non definitivamente uguali a $B-1$, dette *cifre* della rappresentazione;

tali per cui si ha

$$x = \text{sign}(x) B^p \sum_{i=1}^{+\infty} d_i B^{-i}. \tag{2.1}$$

La rappresentazione (2.1) di un numero reale $x$ si dice rappresentazione *normalizzata* in *virgola mobile* (*floating point*) in quanto l'esponente $p$ è determinato in modo da avere parte intera nulla e prima cifra dopo la virgola non nulla.

Si osserva che

- Le condizioni $d_1 \neq 0$ (rappresentazione normalizzata) e $d_i$ non definitivamente uguale a $B-1$ sono introdotte per garantire l'unicità della rappresentazione. A titolo di esempio in base $B = 10$ si ha

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

  ma la seconda non risulta accettabile perchè ha la prima cifra nulla. Analogamente in base $B = 10$ si ha

  $$1 = +10^1(1 \cdot 10^{-1}) = 0.\bar{9} = +10^0 \sum_{i=1}^{+\infty} 9 \cdot 10^{-i}$$

  ma la seconda non risulta accettabile perchè ha le cifre tutte e quindi definitivamente uguali a $9 = 10 - 1$.

- Il numero $x = 0$ non ammette rappresentazione normalizzata. In macchina verrà trattato e memorizzato in modo speciale.

- La rappresentazione floating point dei numeri reali si estende all'insieme dei numeri complessi $z = a + ib$ rappresentati come coppie di numeri reali.

Poichè i registri delle unità aritmetiche ed i dispositivi di memoria di un calcolatore consentono la memorizzazione di un numero finito di cifre binarie *l'insieme dei numeri di macchina*, cioè l'insieme dei numeri reali esattamente rappresentabili in macchina ha cardinalità finita. Risulta pertano essenziale determinare criteri e condizioni che consentano la rappresentazione quanto più accurata possibile dei numeri reali nel calcolatore.

Dal teorema di rappresentazione in base segue che la rappresentazione di un numero reale nel calcolatore può avvenire assegnando delle posizioni di memoria per il segno, per l'esponente e per le cifre della rappresentazione. In tal modo la rappresentazione di un numero in macchina assume la seguente struttura


**Esempio 2.1.1.** I personal computer che implementano lo standard IEEE 754-1985 prevedono la memorizzazione su registri lunghi 32 bit ripartiti come $1 + 8 + 23$ per la *singola precisione* e 64 bit ripartiti come $1 + 11 + 52$ per la *doppia precisione*.

**Definizione 2.1.1.** Si definisce *insieme dei numeri di macchina* in rappresentazione floating point con $t$ cifre, base $B$ e range $(-m, M)$ l'insieme dei numeri reali

$$\mathbb{F}(B, t, m, M) =$$
$$\{0\} \cup \{x \in \mathbb{R} \colon x = \text{sign}(x) B^P \textstyle\sum_{i=1}^{t} d_i B^{-i}, 0 \le d_i \le B-1, d_1 \neq 0, -m \le p \le M\}.$$

Si osserva che

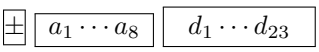
- L'insieme dei numeri di macchina $\mathbb{F}(B, t, m, M)$ ha cardinalità finita $N = 2B^{t-1}(B-1)(M+m+1)+1$;

- Se $x \in \mathbb{F}(B, t, m, M)$ e $x \neq 0$ allora $\omega = B^{-m-1} \le |x| \le B^M(1-B^{-t}) = \Omega$. Ne segue che non è possibile rappresentare esattamente numeri non nulli di modulo minore ad $\omega$. Per aggirare questa limitazione lo standard IEEE754 prevede anche una rappresentazione *denormalizzata*. Quando $p = -m$ la condizione $d_1 \neq 0$ può essere abbandonata e quindi vengono rappresentati numeri positivi e negativi compresi in modulo tra $B^{-m-t}$ e $B^{-m}(B^{-1} - B^{-t})$. Analogamente se $p = M$ si introducono rappresentazioni speciali per i simboli $\pm\infty$ e NaN *not a number*.

- L'insieme dei numeri di macchina $\mathbb{F}(B, t, m, M)$ è simmetrico rispetto all'origine.

- Posto $x = (-1)^s B^p \alpha \in \mathbb{F}(B, t, m, M)$ allora il successivo numero di macchina risulta essere $y = (-1)^s B^p(\alpha + B^{-t})$ per cui in un sistema a virgola mobile la distanza $|y - x| = B^{p-t}$ varia con $p$ ovvero con l'ordine di grandezza dei numeri considerati.

**Esempio 2.1.2.** Si consideri una rappresentazione su 32 bit in base 2 del tipo



In notazione normalizzata la prima cifra $d_0 = 1$ non viene rappresentata. Per l'esponente abbiamo 256 possibili valori che si riducono a 254 eliminando le configurazioni con tutti zero e tutti 1. Mediante una tecnica di traslazione questi 254 valori sono utilizzati per rapprentare gli interi nell'intervallo $[-125, 128]$. Se $a_1 = \ldots = a_8 = 0$ allora il numero rappresentato è $x = \pm(0.0d_1d_2 \ldots d_{23})_2 \cdot 2^{-125}$ (rappresentazione denormalizzata con $d_0 = 0$). Se $a_1 = \ldots = a_8 = 1$ allora $x = \pm\infty$ se $d_1 = \ldots = d_{23} = 0$ e $x =$NaN altrimenti. Per valori dell'esponente compresi tra 1 e 254 il numero rappresentato è $x = \pm(0.1d_1d_2 \ldots d_{23})_2 \cdot 2^{p-126}$ (rappresentazione normalizzata con $d_0 = 1$). Il più piccolo numero positivo normalizzato è $\omega = (1.00\ldots0)_2 \cdot 2^{-126} = 2^{-126}$. Il più grande numero normalizzato è $\Omega = (1.11\ldots1) \cdot 2^{127} = 2^{128}(1 - 2^{-24})$. Per il range si trova $m = 125$ e $M = 128$. Il numero di cifre è $t = 24$. Il primo numero di macchina più grande di $1 = (1.00\ldots0)_2 \cdot 2^0$ è $1 + 2^{-23}$.

# Lezione 2.2: Aritmetica di Macchina.

Rappresentare un numero reale non nullo $x \in \mathbb{R}$, $x \neq 0$, in macchina significa *approssimare* $x$ con un numero $\tilde{x} \in \mathbb{F}(B, t, m, M)$ commettendo un *errore relativo* di rappresentazione

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

quanto più piccolo possibile in valore assoluto. La quantità

$$\eta_x = \tilde{x} - x$$

è detta *errore assoluto* di rappresentazione. In ambito ingegneristico e scientifico l'errore relativo è importante per la valutazione *qualitativa* del fenomeno considerato. Diversamente, un errore assoluto stimato in 1 cm assume differente significato se riferito alla misura di un tavolo o di una distanza astronomica. Il sistema floating point è costruito in modo da assicurare per il modulo dell'errore relativo di rappresentazione (e non per il modulo dell'errore assoluto) una maggiorazione indipendente dalla grandezza del numero rappresentato $x$.

Dato $x \in \mathbb{R}$, $x \neq 0$, distinguiamo 2 casi

1. $|x| < \omega$ (*underflow*) o $|x| > \Omega$ (*overflow*);
2. $\omega \leq |x| \leq \Omega$.

Nel secondo caso le tecniche di approssimazione previste dallo standard IEEE 754 sono:

1. *round to the nearest* (arrotondamento): il numero $x$ viene approssimato con il numero rappresentabile $\tilde{x}$ più vicino;
2. *round toward zero* (troncamento): il numero $x$ viene approssimato con il più grande numero rappresentabile $\tilde{x}$ il cui valore assoluto risulti minore od uguale al valore assoluto di $x$;

3. *round toward plus infinity*: il numero $x$ viene approssimato al più piccolo numero rappresentabile maggiore del dato;

4. *round toward minus infinity*: il numero $x$ viene approssimato al più grande numero rappresentabile minore del dato.

Assumiamo per semplicità di considerare una macchina che opera con troncamento sull'insieme $\mathbb{F}(B, t, m, M)$. Per convenzione indichiamo con $\text{trn}(x) = \tilde{x}$ il risultato dell'approssimazione di $x$ con troncamento e più generalmente $\text{fl}(x)$ l'approssimazione in macchina del dato $x$ nel sistema floating point considerato. Il primo risultato fornisce una maggiorazione uniforme dell'errore di rappresentazione.

**Teorema 2.2.1.** Sia $x \in \mathbb{R}$ con $\omega \leq |x| \leq \Omega$. Si ha

$$|\epsilon_x| = |\frac{\text{trn}(x) - x}{x}| \leq u = B^{1-t}.$$

*Dimostrazione.* Sia $x = (-1)^s B^p \alpha$. L'errore assoluto $|\,\text{trn}(x) - x|$ risulta maggiorato dalla distanza tra due numeri di macchina consecutivi e quindi

$$|\,\text{trn}(x) - x| \leq B^{p-t}.$$

Inoltre $|x| \geq B^{p-1}$. Pertanto vale

$$|\epsilon_x| = |\frac{\text{trn}(x) - x}{x}| \leq \frac{B^{p-t}}{B^{p-1}} = B^{1-t} = u.$$

□

Si osservi che:

- La quantità $u$ detta *precisione di macchina* è indipendente dalla grandezza del numero e caratteristica dell'aritmetica floating point (insieme dei numeri rappresentabili e tecnica di approssimazione) implementata sulla macchina su cui stiamo operando. Se ad esempio operiamo con arrotondamento la distanza tra $x$ e la sua approssimazione di macchina e quindi la precisione di macchina si dimezzano.

- Per valutare la precisione di macchina possiamo determinare il più piccolo numero di macchina maggiore di 1. Detto infatti $x$ tale numero abbiamo che $x - 1 = |x - 1| = B^{1-t}$ essendo $1 = B^1 \cdot B^{-1}$ rappresentato con esponente $p = 1$. Il seguente script MatLab fornisce il valore richiesto

```
eps=0.5;
eps1=eps+1;
while(eps1>1)
eps=0.5*eps;
eps1=eps+1;
end
eps=2*eps
```

- Dal teorema precedente si ricava che dato $x \in \mathbb{R}$ in assenza di situazioni di overflow ed underflow per la sua rappresentazione in macchina vale $\mathrm{fl}(x) = x(1+\epsilon_x)$ con $|\epsilon_x| \leq u$ ed $u$ la precisione di macchina relativa. Questa relazione esprime il modo in cui viene generalmente descritto il legame tra un numero reale e la sua rappresentazione in macchina.

Per le operazioni aritmetiche eseguite in un sistema floating point si pone un analogo problema di approssimazione in quanto il risultato dell'operazione eseguita tra due numeri di macchina in generale non sarà un numero di macchina. Indichiamo con $\oplus, \ominus, \otimes, \oslash$ le operazioni aritmetiche di macchina corrispondenti rispettivamente all'addizione, sottrazione, prodotto e divisione. Richiediamo che le operazioni di macchina siano interne all'insieme dei numeri di macchina e forniscano ovviamente un'approssimazione quanto più accurata possibile del risultato esatto. Una ragionevole definizione risulta pertanto la seguente :

$$\forall a, b \in \mathbb{F}(B, t, m, M), \quad a \oplus b = \mathrm{fl}(a+b),$$

e similmente per le altre operazioni. In tal modo in assenza di situazioni di overflow ed underflow dal teorema precedente segue che

$$\forall a, b \in \mathbb{F}(B, t, m, M), \quad a \oplus b = \mathrm{fl}(a+b) = (a+b)(1+\epsilon_1), \quad |\epsilon_1| \leq u,$$

con $\epsilon_1$ detto *errore locale dell'operazione*.

Se $a, b \in \mathbb{R}$ in assenza di situazioni di overflow ed underflow si ha

$$\mathrm{fl}(a+b) = \mathrm{fl}(a) \oplus \mathrm{fl}(b) = (a(1+\epsilon_a)+b(1+\epsilon_b))(1+\epsilon_1) \doteq (a+b)+a\epsilon_a+b\epsilon_b+(a+b)\epsilon_1,$$

dove con il simbolo $\doteq$ si intende che l'uguaglianza vale considerando le sole componenti lineari negli errori e trascurando le componenti di ordine superiore al primo (sinteticamente riferita come *analisi al primo ordine* dell'errore). Si ottiene quindi che se $a, b \in \mathbb{R}$ , $a+b \neq 0$, in assenza di situazioni di overflow ed underflow vale

$$\frac{\mathrm{fl}(a+b)-(a+b)}{a+b} \doteq \frac{a}{a+b}\epsilon_a + \frac{b}{a+b}\epsilon_b + \epsilon_1,$$

che esprime la dipendenza dell'errore totale commesso nel calcolo della somma tra due numeri reali rispetto agli errori generati dall'approssimazione dei dati iniziali *errore inerente* e agli errori generati dall'algoritmo di calcolo *errore algoritmico* visto come sequenza di operazioni aritmetiche.

Nella prossima lezione tratteremo più in dettaglio questa dipendenza mostrando che la decomposizione è generale e si applica al calcolo di una generica funzione razionale.

# Lezione 2.3: Esercizi.

**Esercizio 1.** Sia $\mathbb{F} = \mathbb{F}(2, 3, 2, 1)$ l'insieme dei numeri di macchina e si supponga di operare con troncamento.

1. Si determini la precisione di macchina.

2. Si determini il minimo numero positivo di macchina $\omega$.

3. Si determini il massimo numero di macchina $\Omega$.
4. Si determini la cardinalità dell'insieme dei numeri di macchina.
5. Si dica se $2/3 \in \mathbb{F}$ e si determini $\mathrm{trn}(2/3)$.
6. Quanti numeri $x \in \mathbb{F}$ soddisfano $1 \leq x < 2$?
7. Quanti numeri $x \in \mathbb{F}$ soddisfano $3/2 \leq x < 2$?
8. Quanti numeri $y \in \mathbb{F}$ soddisfano $1/2 < y \leq 1$?
9. Quanti numeri $y \in \mathbb{F}$ soddisfano $1/2 < y \leq 2/3$?
10. Determinare $x \in \mathbb{F}$ tale che $(1 \oslash x) \otimes x \neq 1$.

# Capitolo 3

# Analisi degli Errori

## Lezione 3.1: Errori nel Calcolo di una Funzione Razionale.

Sia $f\colon [a,b] \to R$ una funzione razionale e $x \in [a,b]$. Un algoritmo per il calcolo di $f(x)$ esprime tale valore come risultato di una sequenza di operazioni aritmetiche. Ad esempio

$$f(x) = \frac{x^2+1}{x} = ((x \cdot x) + 1)/x.$$

Errori nel calcolo di $f(x)$ vengono generati

1. dall'appprossimazione del dato $x$ in macchina con la sua approssimazione $\tilde{x} \in \mathbb{F}(B,t,m,M)$;
2. dall'approssimazione della funzione $f$ in macchina mediante una sua realizzazione $g$, ad esempio $g(x) = ((x \otimes x) \oplus 1) \oslash x$, espressa come corrispondente sequenza di operazioni aritmetiche di macchina.

La prima sorgente di errori conduce alla seguente definizione.

**Definizione 3.1.1.** Si dice *errore inerente* o *inevitabile* generato nel calcolo di $f(x) \neq 0$ la quantità

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

Si osserva che:

- L'errore inerente misura la sensibilità della funzione $f$ e pertanto del *problema matematico* considerato rispetto alla perturbazione del dato iniziale. È indipendente dall'algoritmo (sequenza di operazioni aritmetiche) utilizzato per il calcolo di $f(x)$ e quindi per la risoluzione del problema matematico associato.
- Se l'errore inerente è qualitativamente elevato in valore assoluto diciamo che il relativo problema matematico è *mal condizionato*. Viceversa se l'errore inerente è qualitativamente modesto in valore assoluto diciamo che il

relativo problema matematico è *ben condizionato*. Il termine "qualitativamente" è qui utilizzato per indicare che la valutazione è dipendente dal contesto applicativo.

**Esempio 3.1.1.** Per il calcolo di $f(x) = \frac{x^2+1}{x}$ si ha

$$f(\tilde{x}) = \frac{\tilde{x}^2+1}{\tilde{x}} \doteq \frac{(x^2(1+2\epsilon_x)+1)(1-\epsilon_x)}{x} \doteq \frac{x^2+1}{x} + \epsilon_x(2x - \frac{x^2+1}{x}),$$

e quindi

$$|\epsilon_{in}| = |\frac{f(\tilde{x}) - f(x)}{f(x)}| \doteq |\frac{x^2-1}{x^2+1}||\epsilon_x| \leq u,$$

per cui il problema del calcolo di $f(x)$ risulta ben condizionato.

La seconda sorgente di errori conduce alla seguente definizione.

**Definizione 3.1.2.** Si dice *errore algoritmico* generato nel calcolo di $f(\tilde{x}) \neq 0$ la quantità

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

Si osserva che:

1. La funzione $g$ dipende dall'algoritmo utilizzato per calcolare $f(x)$. Ad esempio per $f(x) = \frac{x^2+1}{x}$ potremmo avere $g(x) = g_1(x) = ((x \otimes x) \oplus 1) \oslash x$ come sopra oppure $g(x) = g_2(x) = x \oplus (1 \oslash x)$. In generale differenti algoritmi conducono a differenti errori algoritmici.

2. Se l'errore algoritmico è qualitativamente elevato in valore assoluto diciamo che l'algoritmo è numericamente *instabile*. Viceversa se l'errore algoritmico è qualitativamente modesto in valore assoluto diciamo che l'algoritmo è numericamente *stabile*.

**Esempio 3.1.2.** Per la valutazione dell'errore algoritmico nel calcolo di $f(x) = \frac{x^2+1}{x}$, $x \in \mathbb{F}(B, t, m, M)$, si considerano le implementazioni $g_1(x)$ e $g_2(x)$. Vale

$$g_1(x) \doteq (x^2(1+\epsilon_1)+1)(1+\epsilon_2)(1+\epsilon_3)/x \doteq \frac{x^2+1}{x} + \frac{x^2+1}{x}(\epsilon_2+\epsilon_3) + x\epsilon_1,$$

da cui

$$|\epsilon_{alg_1}| = |\frac{g_1(x) - f(x)}{f(x)}| \doteq |(\epsilon_2+\epsilon_3) + \frac{x^2}{x^2+1}\epsilon_1| \leq 4u,$$

e pertanto il primo algoritmo risulta numericamente stabile. Riguardo il secondo algoritmo invece si ottiene:

$$g_2(x) = (x + \frac{1+\delta_1}{x})(1+\delta_2) \doteq x + 1/x + (x+1/x)\delta_2 + \delta_1/x,$$

da cui si ricava

$$|\epsilon_{alg_2}| = |\frac{g_2(x) - f(x)}{f(x)}| \doteq |\delta_2 + \delta_1/(x^2+1)| \leq 2u,$$

e pertanto si conclude che anche il secondo algoritmo è numericamente stabile. In altre situazioni la scelta dell'algoritmo di calcolo può risultare critica. Il lettore consideri ad esempio il caso in cui $f(x) = (x-1)/x = 1 - 1/x$.

**Definizione 3.1.3.** Si dice *errore totale* generato nel calcolo di $f(x) \neq 0$ mediante l'algoritmo specificato da $g$ la quantità

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

L'errore totale misura la differenza relativa tra l'output atteso e l'output effettivamente calcolato. In un'analisi al primo ordine vale

**Teorema 3.1.1.** Si ha $\epsilon_{tot} = \epsilon_{in} + \epsilon_{alg}$.

*Dimostrazione.* Vale

$$\epsilon_{tot} = \frac{g(\tilde{x}) - f(x)}{f(x)} = \frac{g(\tilde{x}) - f(\tilde{x})}{f(\tilde{x})}\frac{f(\tilde{x})}{f(x)} + \frac{f(\tilde{x}) - f(x)}{f(x)} =$$
$$= \epsilon_{alg}(1 + \epsilon_{in}) + \epsilon_{in} \doteq \epsilon_{alg} + \epsilon_{in}.$$

□

Il teorema esprime il fatto che nel calcolo di una funzione razionale in un'analisi al primo ordine le due fonti di generazione degli errori individuate precedentemente forniscono contributi separati che possono essere analizzati indipendentemente. L'obiettivo dell'analisi numerica è pertanto quello di individuare algoritmi numericamente stabili per problemi ben condizionati.

# Lezione 3.2: Tecniche per l'Analisi degli Errori.

La regolarità della funzione $f(x)$ ha implicazioni sulle proprietà del problema matematico da essa specificato. La continuità della funzione implica la *buona positura del problema*. Dalla relazione

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

si ricava che la differenziabilità di $f(x)$ è essenziale per il controllo dell'errore inerente. In particolare se assumiamo che $f(x)$ è derivabile due volte con continuità in $(a, b)$ allora vale lo sviluppo di Taylor

$$f(\tilde{x}) = f(x) + f'(x)(\tilde{x} - x) + \frac{f''(\xi)}{2}(\tilde{x} - x)^2, \quad |\xi - x| \leq |\tilde{x} - x|,$$

da cui si ottiene

$$\epsilon_{in} = \frac{f(\tilde{x}) - f(x)}{f(x)} \doteq \frac{f'(x)}{f(x)}x\epsilon_x = c_x\epsilon_x, \quad c_x = \frac{f'(x)}{f(x)}x.$$

La quantità $c_x = c_x(f) = \frac{f'(x)}{f(x)}x$ detta *coefficiente di amplificazione* fornisce una misura del condizionamento del problema. Più generalmente se $f\colon \Omega \to \mathbb{R}$ è definita su un insieme aperto di $\mathbb{R}^n$, differenziabile due volte su $\Omega$ ed il segmento di estremi $\tilde{\boldsymbol{x}}$ e $\boldsymbol{x}$ è contenuto in $\Omega$ allora vale

$$\epsilon_{in} = \frac{f(\tilde{\boldsymbol{x}}) - f(\boldsymbol{x})}{f(\boldsymbol{x})} \doteq \frac{1}{f(\boldsymbol{x})}\sum_{i=1}^{n}\frac{\partial f}{\partial x_i}(\boldsymbol{x})x_i\epsilon_{x_i} = \sum_{i=1}^{n} c_{x_i}(f)\epsilon_{x_i},$$

con

$$c_{x_i}(f) = \frac{1}{f(\boldsymbol{x})}\frac{\partial f}{\partial x_i}(\boldsymbol{x})x_i, \quad 1 \le i \le n,$$

detti coefficienti di amplificazione della funzione $f$ rispetto alla variabile $x_i$.

**Esempio 3.2.1.** Per $f(x) = (x^2+1)/x$ si ha

$$c_x = (2 - (x^2+1)/x^2) \cdot (x/(x^2+1)) \cdot x = (x^2-1)/(x^2+1).$$

Poichè $|c_x| \le 1$ il problema del calcolo di $f(x)$ risulta ben condizionato.

**Esempio 3.2.2.** Per le operazioni aritmetiche si ottiene:

$$\begin{array}{ll}
f(x,y) = x+y, & \epsilon_{in} \doteq c_x\epsilon_x + c_y\epsilon_y, \quad c_x = \dfrac{x}{x+y},\ c_y = \dfrac{y}{x+y};\\
f(x,y) = x-y, & \epsilon_{in} \doteq c_x\epsilon_x + c_y\epsilon_y, \quad c_x = \dfrac{x}{x-y},\ c_y = -\dfrac{y}{x-y};\\
f(x,y) = x\cdot y, & \epsilon_{in} \doteq c_x\epsilon_x + c_y\epsilon_y, \quad c_x = 1,\ c_y = 1;\\
f(x,y) = x/y, & \epsilon_{in} \doteq c_x\epsilon_x + c_y\epsilon_y, \quad c_x = 1,\ c_y = -1.
\end{array}$$

Segue che la sottrazione di due numeri vicini tra loro è potenzialmente causa di elevata amplificazione degli errori relativi (di rappresentazione) cui sono soggetti gli addendi (*fenomeno della cancellazione numerica*). Differentemente le operazioni moltiplicative risultano ben condizionate. Ad esempio siano $x = 0.2178\ 10^2$ e $y = 0.218\ 10^2$ e si supponga di operarare con troncamento in base 10 con 3 cifre di rappresentazione ($u = 10^{-2}$). Si ha $\tilde{x} = 0.217\ 10^2$ e $\tilde{y} = y$. Pertanto $\tilde{x} \ominus \tilde{y} = -0.001\ 10^2 = -0.1$ mentre $x - y = -0.0002\ 10^2 = -0.2\ 10^{-1}$ e quindi $|\epsilon_{in}| = 0.8/0.2 = 0.4$.

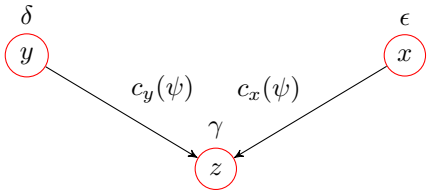
L'analisi dell'errore algoritmico può basarsi sui risultati ottenuti per l'errore inerente. Si distinguono

1. tecniche di analisi *in avanti*;
2. tecniche di analisi *all'indietro*.

Per l'analisi in avanti dell'errore algoritmico si consideri uno step intermedio dell'algoritmo di calcolo ove abbiamo da calcolare il valore dell'operazione aritmetica $\psi(\tilde{x}, \tilde{y})$ a partire da due dati perturbati $\tilde{x} = x(1+\epsilon)$ e $\tilde{y} = y(1+\delta)$. Per quanto visto prima si ha

$$\psi(\tilde{x}, \tilde{y}) \doteq \psi(x,y)(1 + c_x(\psi)\epsilon + + c_y(\psi)\delta + \gamma), \quad |\gamma| \le u,$$

dove $c_x(\psi), c_y(\psi)$ e $\gamma$ sono rispettivamente i coefficienti di amplificazione e l'errore locale dell'operazione $\psi$. Il calcolo dell'errore algoritmico totale può essere reso intuitivo con l'aiuto di un grafo rappresentato nella figura seguente ove i nodi corrispondono ai risultati intermedi generati dall'algoritmo. Se il risultato intermedio $x$ corrisponde al nodo $i$ mentre il risultato intermedio $y$ corrisponde al nodo $j$ allora l'operazione $\psi(x,y) = z$ genera un arco dal nodo $i$ al nodo corrispondente a $z$ con peso $c_x(\psi)$ ed un arco dal nodo $j$ al nodo corrispondente a $z$ con peso $c_y(\psi)$. Nel nodo corrispondente a $z$ viene poi generato un nuovo errore locale $\gamma$ riportato a fianco del nodo. L'errore algoritmico totale accumulato sul nodo corrispondente a $z$ risulta dato dalla somma dell'errore locale più i pesi di arco moltiplicati per l'errore algoritmico totale accumulato sul nodo di origine dell'arco considerato in accordo alla relazione precedente.




L'analisi in avanti dell'errore algoritmico conduce generalmente a valutazioni eccessivamente pessimistiche assumendo l'amplificazione massima ad ogni passo intermedio dell'algoritmo.

Per l'analisi all'indietro dell'errore algoritmico si assume che $g(\tilde{x}) \doteq f(\hat{x})$, ovvero che il valore effettivamente calcolato $g(\tilde{x})$ risulti uguale in un'analisi al primo ordine al valore assunto dalla funzione esatta $f$ valutata in un dato perturbato $\hat{x}$. Se otteniamo una stima sull'errore $(\hat{x}-\tilde{x})/\tilde{x}$ (da qui l'appellativo *all'indietro*) allora siamo in grado di stimare l'errore algoritmico

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

utilizzando i risultati per l'amplificazione dell'errore inerente. L'analisi all'indietro dell'errore algoritmico restituisce generalmente stime più realistiche ed eventualmente se possibile permette di concludere la stabilità dell'algoritmo in situazioni di buon condizionamento del problema. Trova ampia applicazione nell'analisi della stabilità degli algoritmi per l'algebra lineare numerica.

**Esempio 3.2.3.** Si consideri l'algoritmo $g(a,b) = (a \otimes a) \ominus (b \otimes b)$ per il calcolo di $f(a,b) = a^2 - b^2$. Si ha $g(a,b) \doteq a^2(1+\epsilon_1+\epsilon_3) - b^2(1+\epsilon_2+\epsilon_3)$ e quindi $g(a,b) = f(\hat{a},\hat{b})$ con $\hat{a} = a\sqrt{1+\epsilon_1+\epsilon_3} \doteq a(1+\delta_1)$ e $\hat{b} = b\sqrt{1+\epsilon_2+\epsilon_3} \doteq b(1+\delta_2)$ con $|\delta_1| \leq u$ e $|\delta_2| \leq u$. Si confrontino dunque i risultati ottenuti per l'errore algoritmico con l'analisi in avanti (grafo) e l'analisi all'indietro.

## Lezione 3.3: Cenni sul Calcolo di una Funzione non Razionale.

Nel calcolo di una funzione non razionale $h(x)$ si introduce un errore iniziale detto *errore di approssimazione* o *errore analitico* determinato dalla necessità di approssimare la funzione $h(x)$ con una funzione razionale $f(x)$.

**Definizione 3.3.1.** Si dice *errore analitico* generato nel calcolo di $h(x) \neq 0$ mediante la sua approssimazione razionale $f(x)$ la quantità

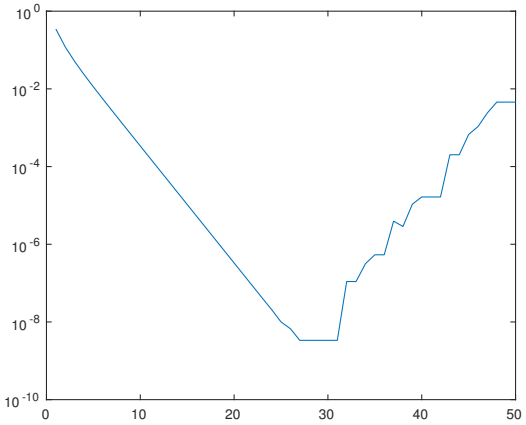
$$\epsilon_{an} = \frac{f(x) - h(x)}{h(x)}.$$

Nel calcolo della funzione $f(x)$ si genera poi un errore inerente ed algoritmico in accordo a quanto visto sopra per cui in un'analisi al primo ordine si perviene alla relazione

$$\epsilon_{tot} = \frac{g(\tilde{x}) - h(x)}{h(x)} \doteq \epsilon_{an} + (\epsilon_{in} + \epsilon_{alg}).$$

Il problema computazionale risulta pertanto quello di determinare approssimazioni razionali che consentano un bilanciamento tra le varie componenti presenti nella stima dell'errore totale.

**Esempio 3.3.1.** Si consideri il problema di approssimare la derivata della funzione $f(x) = \tan(x)$. Per $x$ fissato si pone $g(h) = \frac{f(x+h)-f(x)}{h}$ e si considera l'approssimazione $g(h)$ di $f'(x)$ per valori di $h > 0$ decrescenti. Il seguente grafico riporta il plot dell'errore totale $\epsilon_{tot} = |\frac{\text{fl}(g(h)) - f'(x)}{f'(x)}|$ per $x = 1/3$ e $h = 2^{-k}$, $1 \leq k \leq 50$. L'andamento del grafico rivela che per valori sufficiente-




mente piccoli di $h$ l'errore inerente ed algoritmico commessi nel calcolo di $g(h)$ risultano predominanti rispetto alla riduzione dell'errore analitico. Per esercizio il lettore valuti l'errore inerente nel calcolo di $g(h)$ e l'errore algoritmico connesso nel calcolo di $g(h)$ assunto di disporre di valori calcolati in macchina per $f(x)$ e $f(x+h)$ con errore relativo limitato dalla precisione di macchina.

## Lezione 3.4: Esercizi.

**Esercizio 2.** Si consideri il calcolo della funzione

$$f(a) = \frac{1}{a(1+a)} = \frac{1}{a} - \frac{1}{1+a}, \quad a \neq 0, -1.$$

1. Si studi il condizionamento del problema.
2. Si analizzi la stabilità degli algoritmi di calcolo evidenziati dalle due rappresentazioni della funzione.
3. Si implementino due funzioni per il calcolo della funzione che utlizzano queste rappresentazioni.
4. Si confronti l'output generato da questi programmmi per $a = 1.0e+6$. In particolare si valutino gli errori relativi assumendo come valore esatto

$$f(1.0e+6) = 0.000000000000999999000001000056902485628878 25.$$

5. Si giustifichino i risultati sulla base dell'analisi dell'errore.

**Esercizio 3.** Sia

$$y = f(x) = \frac{(1+x)^2 - (2x+1)}{x^2}, \quad x \neq 0.$$

1. Si calcoli $y = f(x)$ per $x = 10^{-j}$, $1 \leq j \leq 10$.
2. Si determini la sequenza degli errori relativi. Cosa si osserva?
3. Si studi il condizionamento del calcolo di $f(x)$ per $x \neq 0$.
4. Si studi il stabilità del calcolo di $f(x)$ per $x \neq 0$.

**Esercizio 4.** Sia

$$y = f(x_1, \ldots, x_n) = \sum_{i=1}^{n} \frac{1}{x_i}, \quad x_i > 0,\ 1 \leq i \leq n.$$

1. Si studi il condizionamento del calcolo di $f(x_1, \ldots, x_n)$.
2. Si descriva un algoritmo per il calcolo di $y = f(x_1, \ldots, x_n)$ dati $x_1, \ldots, x_n$.
3. Si studi la stabilità dell'algoritmo.
4. Per $n = 10^9$ e $x_i = i^2$ si confrontino i valori ottenuti in MatLab utilizzando due differenti algoritmi di somma ( dal termine più piccolo al più grande e viceversa). (Si assuma come valore esatto il limite della serie $l = \pi^2/6$).

**Esercizio 5.** Si vuole studiare il condizionamento del problema del calcolo dell'integrale definito

$$f \in C^0([0,1]) \to \int_0^1 f(x)dx.$$

Si assuma che la funzione $f$ in macchina sia approssimata mediante la funzione $g \in C^0([0,1])$ tale che

$$g(x) = f(x)(1+\delta_x), \quad |\delta_x| \leq u.$$

1. Si mostri che

$$|\int_0^1 f(x)dx - \int_0^1 g(x)dx| \leq u \int_0^1 |f(x)|dx.$$

2. Si mostri che

$$\frac{|\int_0^1 f(x)dx - \int_0^1 g(x)dx|}{|\int_0^1 f(x)dx|} \leq u \frac{\int_0^1 |f(x)|dx}{|\int_0^1 f(x)dx|}.$$

3. Si consideri l'approssimazione dell'integrale ($n$ pari)

$$I_n = \int_0^1 sin(n\pi x)e^{-x}dx = \frac{n\pi(1 + ((-1)^{n+1})/e)}{1+n^2\pi^2}.$$

   Si determini un'approssimazione utilizzando la funzione *integral* in MatLab e si discutano i risultati ottenuti.

**Esercizio 6.** Considerare la seguente procedura per valutare sperimentalmente

$$\lim_{h\to 0} \frac{e^h - 1}{h}.$$

Calcolare

$$d_n = \frac{e^{2^{-n}} - 1}{2^{-n}}, \quad n = 0, 1, 2, \ldots,$$

ed accettare come valore del limite il primo termine della successione per cui $d_n = d_{n+1}$.

1. Implementare la procedura.
2. Riportare il valore approssimato del limite ed il valore di $n$ che soddisfa la condizione di arresto.
3. Utilizzando lo sviluppo di Taylor dell'esponenziale nell'origine stimare teoricamente il valore di $n$.

**Esercizio 7.** Scrivere un programma che dato in input il valore di $N \in \mathbb{N}$ restituisce in output il valore di $S_N$

$$S_N = \sum_{n=1}^{N} \left( \frac{1}{n} - \frac{1}{n+1} \right) = \sum_{n=1}^{N} \frac{1}{n(n+1)}$$

calcolato mediante le due formule descritte.

1. Riportare i valori restituiti per $N = 10^k$, $1 \le k \le 7$.
2. Calcolare i corrispondenti errori relativi ed assoluti.
3. Commentare i risultati.

**Esercizio 8.** Posto $a = -0.01$ e $b = 0.01$ si generi il vettore $\boldsymbol{x} = [x_1, \ldots, x_{100}] \in \mathbb{R}^{100}$ di punti equispaziati nell'intervallo $[a, b]$. Sia inoltre $\boldsymbol{f} = [f(x_1), \ldots, f(x_{100})]$ e $\boldsymbol{g} = [g(x_1), \ldots, g(x_{100})]$ con

$$f(x) = \frac{1 - \cos^2 x}{x^2}, \quad g(x) = \frac{\sin^2 x}{x^2}.$$

Sia infine

$$\boldsymbol{w} = [w_1, \ldots, w_{100}], \quad w_j = \frac{|f_j - g_j|}{|g_j|}, \ 1 \le j \le 100.$$

Utilizzando la funzione `semilogy` si rappresenti in scala logaritmica il vettore $\boldsymbol{w}$ e si commenti il risultato.

**Esercizio 9.** Sia

$$y = f(x) = \left(1 + \frac{1}{x}\right)^x, \quad x > 0.$$

1. Si calcoli $\lim_{x\to+\infty} f(x)$.
2. Utilizzando MatLab si calcoli $y = f(x)$ per $x = 10^j$, $10 \le j \le 16$.
3. Si studi il condizionamento del calcolo di $f(x)$ per $x \to +\infty$.
4. Si giustifichino i risultati numerici ottenuti.

# Capitolo 4

# I Problemi dell'Algebra Lineare Numerica: Aspetti Computazionali e Condizionamento

## Lezione 4.1: Norme Matriciali e Norme Vettoriali.

I principali problemi dell'algebra lineare numerica concernono la risoluzione di sistemi lineari ed il calcolo di autovalori e/o autovettori di matrici. Studiarne il condizionamento significa misurare la sensibilità del problema considerato rispetto a perturbazioni dei dati forniti in ingresso. Risulta pertanto essenziale disporre di strumenti per valutare la distanza tra vettori e matrici. La risoluzione di sistemi lineari per dati in ingresso reali può essere eseguita in aritmetica reale. Differentemente gli autovalori/autovettori di una matrice reale possono essere complessi. È opportuno quindi disporre di strumenti che operano su spazi vettoriali $\mathbb{F}^n$ e $\mathbb{F}^{n\times n}$, $n \geq 1$, con $\mathbb{F} \in \{\mathbb{R}, \mathbb{C}\}$.

**Definizione 4.1.1.** Si dice *norma vettoriale* su $\mathbb{F}^n$ una funzione $f\colon \mathbb{F}^n \to \mathbb{R}$ che soddisfa le seguenti proprietà:

1. $\forall \boldsymbol{v} \in \mathbb{F}^n, \quad f(\boldsymbol{v}) \geq 0$ ed inoltre $f(\boldsymbol{v}) = 0 \iff \boldsymbol{v} = \boldsymbol{0}$;
2. $\forall \boldsymbol{v} \in \mathbb{F}^n, \forall \alpha \in \mathbb{F}, \;\; f(\alpha \boldsymbol{v}) = |\alpha| f(\boldsymbol{v})$;
3. $\forall \boldsymbol{v}, \boldsymbol{z} \in \mathbb{F}^n, \;\; f(\boldsymbol{v} + \boldsymbol{z}) \leq f(\boldsymbol{v}) + f(\boldsymbol{z})$.

Se $f$ è una norma vettoriale su $\mathbb{F}^n$ indicheremo per comodità di notazione $f(\boldsymbol{v}) = \| \boldsymbol{v} \|$.

Si osservi che:

- Una norma vettoriale su $\mathbb{F}^n$ induce una *distanza* $d\colon \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{R}$ tra elementi (punti) di $\mathbb{F}^n$ definita come

$$\forall \boldsymbol{v}, \boldsymbol{z} \in \mathbb{F}^n, \quad d(\boldsymbol{v}, \boldsymbol{z}) = \| \boldsymbol{v} - \boldsymbol{z} \| .$$

Le proprietà della norma si traducono in analoghe proprietà della distanza indotta:

1. (*non negatività*) $\forall \boldsymbol{v}, \boldsymbol{z} \in \mathbb{F}^n,\ d(\boldsymbol{v}, \boldsymbol{z}) \geq 0$ e $d(\boldsymbol{v}, \boldsymbol{z}) = 0 \iff \boldsymbol{v} = \boldsymbol{z}$;
2. (*simmetria*) $\forall \boldsymbol{v}, \boldsymbol{z} \in \mathbb{F}^n,\ d(\boldsymbol{v}, \boldsymbol{z}) = d(\boldsymbol{z}, \boldsymbol{v})$;
3. (*diseguaglianza triangolare*) $\forall \boldsymbol{v}, \boldsymbol{z}, \boldsymbol{w} \in \mathbb{F}^n,\ d(\boldsymbol{v}, \boldsymbol{z}) \leq d(\boldsymbol{v}, \boldsymbol{w}) + d(\boldsymbol{w}, \boldsymbol{z})$.

- Le seguenti funzioni $f(\boldsymbol{v})$, $\boldsymbol{v} = [v_1, \ldots, v_n]^T \in \mathbb{R}^n$ sono norme su $\mathbb{R}^n$ (rispettivamente dette *norma euclidea*, *norma 1* e *norma infinito*):

$$\begin{array}{l} f(\boldsymbol{v}) = \| \boldsymbol{v} \|_2 = \sqrt{\sum_{i=1}^n v_i^2} = \sqrt{\boldsymbol{v}^T \boldsymbol{v}}; \\ f(\boldsymbol{v}) = \| \boldsymbol{v} \|_1 = \sum_{i=1}^n |v_i|; \\ f(\boldsymbol{v}) = \| \boldsymbol{v} \|_\infty = \max_{1 \leq i \leq n} |v_i| \end{array}$$

- Le seguenti funzioni $f(\boldsymbol{v})$, $\boldsymbol{v} = [v_1, \ldots, v_n]^T \in \mathbb{C}^n$ sono norme su $\mathbb{C}^n$ (rispettivamente dette *norma euclidea*, *norma 1* e *norma infinito*):

$$\begin{array}{l} f(\boldsymbol{v}) = \| \boldsymbol{v} \|_2 = \sqrt{\sum_{i=1}^n |v_i|^2} = \sqrt{\boldsymbol{v}^H \boldsymbol{v}}; \\ f(\boldsymbol{v}) = \| \boldsymbol{v} \|_1 = \sum_{i=1}^n |v_i|; \\ f(\boldsymbol{v}) = \| \boldsymbol{v} \|_\infty = \max_{1 \leq i \leq n} |v_i| \end{array}$$

- Sebbene l'utilizzo di differenti norme conduca a risultati quantitativamente differenti le proprietà qualitative degli oggetti e dei fenomeni analizzati risultano preservate. Vale infatti il seguente *Principio di equivalenza topologica (metrica) delle norme*:

  **Teorema 4.1.1.** Siano $\| \cdot \|$ e $\| \cdot \|'$ due norme su $\mathbb{F}^n$ . Allora esistono costanti $\alpha, \beta > 0$ tali che:

$$\alpha \| \boldsymbol{v} \|' \leq \| \boldsymbol{v} \| \leq \beta \| \boldsymbol{v} \|', \quad \forall \boldsymbol{v} \in \mathbb{F}^n.$$

  In particolare questo risultato implica che le proprietà (topologiche) di convergenza/divergenza di successioni e di continuità delle funzioni sono invarianti rispetto alla norma considerata.

  ♠ FAC Per una dimostrazione del risultato dalla diseguaglianza triangolare segue che

$$| \| \boldsymbol{x} \| - \| \boldsymbol{y} \| | \leq \| \boldsymbol{x} - \boldsymbol{y} \|, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}^n.$$

  Ciò implica la continuità della funzione $\boldsymbol{x} \to \| \boldsymbol{x} \|$, $\boldsymbol{x} \in \mathbb{F}^n$. Dal teorema di Weierstrass si ricava quindi che la funzione valutata sull'insieme $\mathcal{S} = \{\boldsymbol{v} \in \mathbb{F}^n : \ \| \boldsymbol{v} \|' = 1\}$ ammette massimo detto $\beta$ e minimo detto $\alpha$.

Le norme vettoriali possono essere estese alle matrici.

**Definizione 4.1.2.** Si dice *norma matriciale* su $\mathbb{F}^{n \times n}$ una funzione $f : \mathbb{F}^{n \times n} \to \mathbb{R}$ che soddisfa le seguenti proprietà:

1. $\forall A \in \mathbb{F}^{n \times n},\quad f(A) \geq 0$ ed inoltre $f(A) = 0 \iff A = 0$;

2. $\forall A \in \mathbb{F}^{n\times n}, \forall \alpha \in \mathbb{F},\ \ f(\alpha A) = |\alpha| f(A)$;

3. $\forall A, B \in \mathbb{F}^{n\times n},\ \ f(A+B) \leq f(A) + f(B)$;

4. $\forall A, B \in \mathbb{F}^{n\times n},\ \ f(A\cdot B) \leq f(A)\cdot f(B)$.

Se $f$ è una norma vettoriale su $\mathbb{F}^{n\times n}$ indicheremo per comodità di notazione $f(A) = \| A \|$.

Si osserva che:

- Analogamente a sopra una norma matriciale su $\mathbb{F}^{n\times n}$ induce una *distanza* $d\colon \mathbb{F}^{n\times n} \times \mathbb{F}^{n\times n} \to \mathbb{R}$ tra elementi di $\mathbb{F}^{n\times n}$ definita come

$$\forall A, B \in \mathbb{F}^{n\times n}, \quad d(A,B) = \| A - B \| .$$

- Dalle proprietà 1) e 4) si ricava che $\| I_n \| \geq 1$.

La seguente definizione descrive l'estensione di una norma vettoriale ad una norma matriciale detta *norma matriciale indotta o compatibile* con la norma vettoriale.

**Definizione 4.1.3.** Data $\| \cdot \|$ una norma vettoriale su $\mathbb{F}^n$ si dice *norma matriciale indotta o compatibile* con la norma vettoriale la funzione $f\colon \mathbb{F}^{n\times n} \to \mathbb{R}$ definita da

$$\forall A \in \mathbb{F}^{n\times n}, \quad f(A) = \max_{\{\boldsymbol{v}\in\mathbb{F}^n\colon \|\boldsymbol{v}\|=1\}} \| A\boldsymbol{v} \| .$$

Per comodità di notazione scriveremo $f(A) = \| A \|$ utilizzando lo stesso simbolo della norma vettoriale che induce la norma matriciale.

Si osserva che:

- La definizione è ben posta.

> ♠ FAC Si può infatti dimostrare come sopra che data $A \in \mathbb{F}^{n\times n}$ la funzione $\boldsymbol{v} \to \| A\boldsymbol{v} \|$ è continua per cui dal teorema di Weierstrass esiste il massimo. Inoltre la funzione $f$ così definita verifica le proprietà 1–4 delle norme matriciali.

- Per una norma matriciale indotta da una norma vettoriale vale $\| I_n \| = 1$.

- Esistono norme matriciali che non sono indotte da una norma vettoriale. Ad esempio si consideri la funzione *norma di Frobenius* definita come

$$\forall A = (a_{i,j}) \in \mathbb{F}^{n\times n}, \quad \| A \| = \sqrt{\sum_{1\leq i,j\leq n} |a_{i,j}|^2} = \sqrt{\text{traccia}(A^H A)}.$$

Il seguente risultato esprime un'ulteriore proprietà di fondamentale importanza per l'analisi dei problemi e dei metodi computazionali.

**Teorema 4.1.2.** Sia $\| \cdot \|$ una norma vettoriale su $\mathbb{F}^n$ e sia $\| \cdot \|$ la norma matriciale indotta. Vale allora che

$$\forall A \in \mathbb{F}^{n\times n},\ \forall \boldsymbol{v} \in \mathbb{F}^n, \quad \| A\boldsymbol{v} \| \leq \| A \| \| \boldsymbol{v} \| .$$

*Dimostrazione.* Se $\boldsymbol{v} = \boldsymbol{0}$ allora la relazione vale. Assumiamo pertanto che $\boldsymbol{v} \neq \boldsymbol{0}$. Si ha

$$\| A\frac{\boldsymbol{v}}{\| \boldsymbol{v} \|} \| \leq \| A \| = \max_{\{\boldsymbol{z} \in \mathbb{F}^n : \| \boldsymbol{z} \| = 1\}} \| A\boldsymbol{z} \|,$$

e quindi la tesi segue per la proprietà 2) delle norme vettoriali. □

La valutazione delle norme matriciali indotte dalla norma euclidea, dalla norma 1 e dalla norma infinito in accordo alla definizione risulta computazionalmente non praticabile. Vengono pertanto fornite le seguenti caratterizzazioni che seguono dalla definizione e mediante l'individuazione del punto di massimo forniscono lo strumento per il calcolo effettivo delle norme.

**Teorema 4.1.3.** Sia $A = (a_{i,j}) \in \mathbb{F}^{n \times n}$. Si ha

$$\| A \|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{i,j}|;$$
$$\| A \|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{i,j}|.$$

Detto inoltre $\rho(B)$ il *raggio spettrale* di una matrice $B \in \mathbb{F}^{n \times n}$ definito come il modulo dell'autovalore di modulo massimo di $B$, i.e.,

$$\forall B \in \mathbb{F}^{n \times n}, \quad \rho(B) = \max_{1 \leq i \leq n} |\lambda_i|, \quad \lambda_i, 1 \leq i \leq n, \text{ autovalore di } B,$$

allora vale

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

Si osservi che mentre per la norma infinito e la norma 1 il calcolo si realizza facilmente con una sequenza finita di operazioni aritmetiche e confronti a partire dagli elementi della matrice, per la norma euclidea si richiede la risoluzione di un problema ausiliario (il calcolo del modulo dell'autovalore dominante di una matrice associata) di assai maggiore difficoltà (vedi Lezione 4 successiva).

# Lezione 4.2: Il Problema della Risoluzione di un Sistema Lineare ed il suo Condizionamento.

Il problema della risoluzione di un sistema lineare ad elementi reali viene formulato come segue.

**Problema 1.** Data $A = (a_{i,j}) \in \mathbb{R}^{n \times n}$ matrice invertibile e dato $\boldsymbol{b} \in \mathbb{R}^n$ si cerca un vettore $\boldsymbol{x} \in \mathbb{R}^n$ tale che

$$A\boldsymbol{x} = \boldsymbol{b}. \tag{4.1}$$

La matrice $A$ è detta *matrice dei coefficienti* del sistema lineare. Il vettore $\boldsymbol{b}$ è detto *vettore dei termini noti*. Il vettore $\boldsymbol{x}$ è detto *vettore delle incognite*. La scrittura (4.1) è formalmente equivalente al sistema di equazioni lineari

$$\begin{cases} \sum_{j=1}^n a_{1,j} x_j = b_1 \\ \sum_{j=1}^n a_{2,j} x_j = b_2 \\ \vdots \\ \sum_{j=1}^n a_{n,j} x_j = b_n \end{cases}$$

Dall'invertibilità della matrice dei coefficienti segue l'esistenza di un tale $\boldsymbol{x} \in \mathbb{R}^n$ (ad esempio $\boldsymbol{x} = A^{-1}\boldsymbol{b}$) e la sua unicità (altrimenti si avrebbe $\dim(\ker(A)) > 0$). Inoltre la *regola di Cramer* fornisce una descrizione delle componenti del vettore $\boldsymbol{x}$ come funzioni razionali negli elementi della matrice $A$ e del vettore $\boldsymbol{b}$. Quindi in linea di principio è possibile studiare formalmente il condizionamento del problema della risoluzione di un sistema lineare con gli strumenti (differenziali e sviluppi al primo ordine) visti in precedenza. Un tale studio tuttavia risulta poco informativo non evidenziando una misura generale "facilmente" calcolabile del condizionamento del problema in oggetto. Differentemente l'approccio descritto di seguito fornisce una maggiorazione dell'errore imputabile nel risultato alla perturbazione dei dati iniziali $(A, \boldsymbol{b})$ in termini di un parametro detto *numero di condizione o di condizionamento* del sistema lineare che funge in questo contesto da analogo dei coefficienti di amplificazione.

La memorizzazione in macchina dei dati in ingresso $(A, \boldsymbol{b})$ conduce a dati perturbati $(\hat{A}, \hat{\boldsymbol{b}})$ con

$$\hat{A} = (\hat{a}_{i,j}), \quad \hat{a}_{i,j} = a_{i,j}(1 + \epsilon_{i,j}), \quad |\epsilon_{i,j}| \leq u,$$

$$\hat{\boldsymbol{b}} = \left[\hat{b}_1, \ldots, \hat{b}_n\right]^T, \quad \hat{b}_j = b_j(1 + \delta_j), \quad |\delta_j| \leq u.$$

Si può dunque scrivere

$$\hat{A} = A + F, \quad F = (a_{i,j}\epsilon_{i,j}); \quad \hat{\boldsymbol{b}} = \boldsymbol{b} + \boldsymbol{e}, \quad \boldsymbol{e} = [b_1\delta_1, \ldots, b_n\delta_n]^T.$$

Per semplicità di analisi assumiamo di perturbare il solo termine noto $\boldsymbol{b} \neq \boldsymbol{0}$. Le stime così ottenute si estendono facilmente al caso più generale in cui si assumuno perturbazioni sia su $A$ che su $\boldsymbol{b}$. Sia $\hat{\boldsymbol{x}}$ la soluzione del sistema perturbato, i.e.,

$$A\hat{\boldsymbol{x}} = \hat{\boldsymbol{b}}.$$

Si ha

$$\hat{\boldsymbol{x}} - \boldsymbol{x} = A^{-1}\hat{\boldsymbol{b}} - A^{-1}\boldsymbol{b} = A^{-1}\boldsymbol{e},$$

e dunque passando ad una valutazione in norma

$$\| \hat{\boldsymbol{x}} - \boldsymbol{x} \| = \| A^{-1}\boldsymbol{e} \| \leq \| A^{-1} \| \| \boldsymbol{e} \|,$$

ove la norma matriciale è quella indotta dalla norma vettoriale iniziale. D'altra parte abbiamo che

$$\| \boldsymbol{b} \| = \| A\boldsymbol{x} \| \leq \| A \| \| \boldsymbol{x} \|,$$

da cui

$$\| \boldsymbol{x} \| \geq \| \boldsymbol{b} \| \, / \, \| A \| .$$

Combinando insieme queste relazioni si ottiene

$$\frac{\| \hat{\boldsymbol{x}} - \boldsymbol{x} \|}{\| \boldsymbol{x} \|} \leq \mathcal{K}(A) \frac{\| \boldsymbol{e} \|}{\| \boldsymbol{b} \|}, \quad \mathcal{K}(A) = \| A \| \| A^{-1} \| .$$

Questa relazione esprime il fatto che l'errore relativo in norma imputato alla perturbazione dei dati iniziali si maggiora con l'errore relativo sui dati moltiplicato per il parametro $\mathcal{K}(A)$ detto *numero di condizione o di condizionamento*

del sistema lineare che assume il ruolo di coefficiente di amplificazione. Per una norma matriciale indotta da una norma vettoriale vale

$$1 = \| I_n \| = \| AA^{-1} \| \leq \| A \| \| A^{-1} \| = \mathcal{K}(A).$$

Se $\mathcal{K}(A)$ è qualitativamente elevato il sistema lineare è detto *mal condizionato*. Se altrimenti $\mathcal{K}(A)$ è qualitativamente modesto il sistema lineare è detto *ben condizionato*.

In norma infinito si ha

$$\| \boldsymbol{e} \|_\infty \leq u \| \boldsymbol{b} \|_\infty,$$

da cui

$$\frac{\| \hat{\boldsymbol{x}} - \boldsymbol{x} \|_\infty}{\| \boldsymbol{x} \|_\infty} \leq u\mathcal{K}_\infty(A), \quad \mathcal{K}_\infty(A) = \| A \|_\infty \| A^{-1} \|_\infty .$$

Ne ricaviamo che una stima di $\mathcal{K}_\infty(A)$ fornisce un'indicazione sul numero di cifre al più corrette attese sul risultato.

# Lezione 4.3: Il Problema del Calcolo degli Autovalori di una Matrice ed il suo Condizionamento.

Il problema del calcolo di un autovalore di una matrice viene formulato come segue.

**Problema 2.** Data $A = (a_{i,j}) \in \mathbb{C}^{n \times n}$ si cerca $\lambda \in \mathbb{C}$ tale per cui

$$\exists \boldsymbol{x} \in \mathbb{C}^n, \ \boldsymbol{x} \neq \boldsymbol{0} \ : \ A\boldsymbol{x} = \lambda \boldsymbol{x}, \tag{4.2}$$

o, equivalentemente,

$$\exists \boldsymbol{y} \in \mathbb{C}^n, \ \boldsymbol{y} \neq \boldsymbol{0} \ : \ \boldsymbol{y}^H A = \lambda \boldsymbol{y}^H. \tag{4.3}$$

Il vettore $\boldsymbol{x}$ è detto *autovettore (destro)* relativo all'*autovalore* $\lambda$ di $A$. Il vettore $\boldsymbol{y}$ è detto *autovettore (sinistro)* relativo all'*autovalore* $\lambda$ di $A$. Dalla relazione (4.2) o (4.3) segue che $\lambda$ è autovalore di $A$ se e soltanto se

$$\dim(\ker(A - \lambda I_n)) > 0$$

o, equivalentemente, se e soltanto se

$$\det(A - \lambda I_n) = 0.$$

Introdotto pertanto il polinomio

$$p(x) = \det(A - xI_n)$$

detto *polinomio caratteristico* della matrice $A$ segue che $\lambda \in \mathbb{C}$ è autovalore di $A$ se e soltanto se

$$p(\lambda) = \det(A - \lambda I_n) = 0,$$

ovvero $\lambda$ è una radice dell'equazione $p(x) = 0$.

La caratterizzazione degli autovalori come zeri del polinomio caratteristico ha importanti implicazioni teoriche e computazionali. Dal *teorema fondamentale dell'algebra* sappiamo che l'equazione algebrica $p(x) = 0$ ammette esattamente $n$ radici contate con la loro molteplicità. Dette $\lambda_1, \lambda_2, \ldots, \lambda_n$ queste radici (e quindi gli autovalori di $A$) possiamo scrivere la fattorizzazione ottenuta raggruppando le radici uguali nella forma

$$p(x) = (-1)^n \prod_{i=1}^{k} (x - \lambda_{j_i})^{\sigma_i},$$

con $1 \leq j_i \leq n$, $\lambda_{j_i} \neq \lambda_{j_\ell}$ se $i \neq \ell$, $\sum_{i=1}^{k} \sigma_i = n$. Il numero $\sigma_i$ è detto *molteplicità algebrica* dell'autovalore $\lambda_{j_i}$ e indica il numero di volte che $\lambda_{j_i}$ compare nell'insieme $\mathcal{S} = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ detto *spettro* di $A$. Il numero

$$\tau_i = \dim(\ker(A - \lambda_{j_i} I_n)), \quad 1 \leq i \leq k,$$

è detto *molteplicità geometrica* dell'autovalore $\lambda_{j_i}$ e indica il numero di autovettori destri o sinistri linearmente indipendenti relativi all'autovalore $\lambda_{j_i}$. È noto che

$$\sigma_i \geq \tau_i, \quad 1 \leq i \leq k,$$

e la situazione in cui vale l'uguaglianza tra i due indici per ogni $i$ risulta di particolare interesse.

**Definizione 4.3.1.** Sia $A \in \mathbb{C}^{n \times n}$ e $V \in \mathbb{C}^{n \times n}$ invertibile allora la trasformazione $A \to B = V^{-1}AV$ è detta *trasformazione per similitudine* di $A$ mediante $V$.

Nel contesto del calcolo degli autovalori di una matrice le traformazioni per similitudine sono rilevanti in quanto preservano gli autovalori insieme agli indici associati. Vale infatti che

$$\begin{aligned} &\det(B - xI_n) = \det(V^{-1}AV - xI_n) = \det(V^{-1}AV - xV^{-1}V) = \\ &\det[V^{-1}(A - xI_n)V] = \det(V^{-1})\det(A - xI_n)\det(V) = \det((A - xI_n), \end{aligned}$$

e quindi $A$ e $B$ condividono lo stesso polinomio caratteristico. Analogamente si mostra che

$$\dim(\ker(A - \lambda_{j_i} I_n)) = \dim(\ker(B - \lambda_{j_i} I_n)), \quad 1 \leq i \leq k.$$

**Definizione 4.3.2.** La matrice $A \in \mathbb{C}^{n \times n}$ è detta *diagonalizzabile* se esiste una trasformazione per similitudine che rende $A$ diagonale, ovvero,

$$\exists\, V,\ \det(V) \neq 0 \colon V^{-1}AV = D \ \text{ diagonale.}$$

Segue facilmente che se $A$ è diagonalizzabile allora $D = \mathrm{diag}[\lambda_1, \ldots, \lambda_n]$ ed inoltre $AV = VD$ implica che le colonne di $V$ definiscono gli autovettori (destri) corrispondenti. Il seguente teorema fornisce una condizione necessaria e sufficiente per la diagonalizzabilità di una matrice.

**Teorema 4.3.1.** $A \in \mathbb{C}^{n \times n}$ è diagonalizzabile se e soltanto se $\sigma_i = \tau_i$, $1 \leq i \leq k$.

Le seguenti classi di matrici diagonalizzabili risultano di interesse:

1. *Matrici simmetriche.* Una matrice $A \in \mathbb{R}^{n\times n}$ si dice simmetrica se $A = A^T$. Le matrici simmetriche sono diagonalizzabili e la matrice $V$ può essere scelta ortonormale, i.e., $V^TV = VV^T = I_n$. Gli autovalori di una matrice simmetrica sono reali.

2. *Matrici hermitiane.* Una matrice $A \in \mathbb{C}^{n\times n}$ si dice hermitiana se $A = A^H$. Le matrici hermitiane sono diagonalizzabili e la matrice $V$ può essere scelta ortonormale, i.e., $V^HV = VV^H = I_n$. Gli autovalori di una matrice hermitiana sono reali.

3. *Matrici con autovalori distinti.* Qualora $\lambda_i \neq \lambda_j$ se $i \neq j$ allora $k = n$, $\sigma_i = 1$, $1 \leq i \leq n$ e dunque $\tau_i = \sigma_i$, $1 \leq i \leq n$. Un autovalore $\lambda$ per cui $\sigma = \tau = 1$ è detto *semplice*.

Lo studio del condizionamento del calcolo degli autovalori risulta semplificato nell'ipotesi di autovalori semplici. In particolare si assuma che $\lambda$ è autovalore di $A$ con $\sigma = \tau = 1$ e autovettore destro e sinistro denotati rispettivamente con $\boldsymbol{x}$ e $\boldsymbol{y}$. Si consideri una perturbazione $\hat{A}$ di $A$ della forma $\hat{A} = A + \epsilon F$. Per $\epsilon$ sufficientemente piccolo si può dimostrare che $\hat{A}$ ammette un autovalore $\lambda(\epsilon) \doteq \lambda + \epsilon\eta$ con corrispondente autovettore $\boldsymbol{x}(\epsilon) \doteq \boldsymbol{x} + \epsilon\boldsymbol{z}$. Dalla relazione

$$(A + \epsilon F)(\boldsymbol{x} + \epsilon\boldsymbol{z}) \doteq (\lambda + \epsilon\eta)(\boldsymbol{x} + \epsilon\boldsymbol{z}),$$

segue

$$A\boldsymbol{z} + F\boldsymbol{x} \doteq \lambda\boldsymbol{z} + \eta\boldsymbol{x},$$

che moltiplicando ambo i membri per $\boldsymbol{y}^H$ implica

$$\eta \doteq \frac{\boldsymbol{y}^H F\boldsymbol{x}}{\boldsymbol{y}^H\boldsymbol{x}}$$

da cui la stima

$$|\lambda(\epsilon) - \lambda| \doteq |\epsilon||\frac{\boldsymbol{y}^H F\boldsymbol{x}}{\boldsymbol{y}^H\boldsymbol{x}}|.$$

Dalla diseguaglianza di Cauchy–Schwarz si ha che

$$|\boldsymbol{y}^H F\boldsymbol{x}| \leq \| \boldsymbol{y} \|_2 \| F\boldsymbol{x} \|_2 \leq \| \boldsymbol{y} \|_2 \| \boldsymbol{x} \|_2 \| F \|_2$$

e quindi considerando autovettori normalizzati ($\| \boldsymbol{y} \|_2 = \| \boldsymbol{x} \|_2 = 1$) si perviene alla relazione

$$|\lambda(\epsilon) - \lambda| \doteq |\epsilon||\frac{\boldsymbol{y}^H F\boldsymbol{x}}{\boldsymbol{y}^H\boldsymbol{x}}| \leq \| \epsilon F \|_2 \frac{1}{|\boldsymbol{y}^H\boldsymbol{x}|}.$$

Questa relazione indica che il condizionamento (rispetto all'errore assoluto) del calcolo di un autovalore semplice di una matrice è misurato dal reciproco del valore assoluto del prodotto scalare tra i rispettivi autovettori destro e sinistro normalizzati. Il lettore ne derivi che il problema del calcolo di un autovalore semplice di una matrice simmetrica o hermitiana è ben condizionato.

# Lezione 4.4: Teoremi di Localizzazione per Autovalori.

Gli aspetti computazionali della caratterizzazione degli autovalori di una matrice come soluzioni dell'equazione caratteristica associata sono pure rilevanti.

Il teorema di Abel-Ruffini afferma che non esiste nessuna formula per le radici di una generica equazione algebrica di grado $\geq 5$ in funzione dei coefficienti del polinomio, usando solo le operazioni aritmetiche e l'applicazione di radicali (radici quadrate, radici cubiche, ecc.). Ne consegue che, a differenza della soluzione del sistema lineare, gli autovalori di una matrice non sono generalmente esprimibili come funzioni razionali negli elementi della matrice. Pertanto la loro approssimazione si baserà sulla costruzione iterativa di successioni $\{a_k\}_{k\in\mathbb{N}}$ di approssimanti che sotto ipotesi convenienti convergeranno ad un autovalore. Per la convergenza risulta essenziale disporre di un'approssimazione iniziale "sufficientemente buona". I teoremi di localizzazione individuano regioni del piano complesso ove gli autovalori sono confinati. Il più classico è il seguente detto *teorema di Gershgorin*.

**Teorema 4.4.1.** Sia $A = (a_{i,j}) \in \mathbb{C}^{n\times n}$. Definiamo i *cerchi di Gershgorin* $K_i$, $1 \leq i \leq n$, come

$$K_i = \{z \in \mathbb{C}\colon |z - a_{i,i}| \leq \sum_{j=1, j\neq i}^{n} |a_{i,j}|\}, \quad 1 \leq i \leq n.$$

Allora

$$\lambda \text{ autovalore di } A \Rightarrow \lambda \in \cup_{i=1}^{n} K_i.$$

*Dimostrazione.* Sia $\lambda$ autovalore di $A$ con corrispondente autovettore destro $\boldsymbol{x}$. La relazione $A\boldsymbol{x} = \lambda\boldsymbol{x}$ implica che

$$\sum_{j=1}^{n} a_{i,j}x_j = \lambda x_i, \quad 1 \leq i \leq n,$$

da cui

$$(\lambda - a_{i,i})x_i = \sum_{j=1, j\neq i}^{n} a_{i,j}x_j, \quad 1 \leq i \leq n. \tag{4.4}$$

Sia $p$ l'indice di una componente di modulo massimo di $\boldsymbol{x}$, i.e., $|x_p| = \| \boldsymbol{x} \|_\infty$. Poichè $\boldsymbol{x} \neq 0$ si ha $|x_p| > 0$. La relazione per $i = p$ in (4.4) porge

$$(\lambda - a_{p,p})x_p = \sum_{j=1, j\neq p}^{n} a_{p,j}x_j$$

da cui passando ai valori assoluti

$$|(\lambda - a_{p,p})x_p| = |\lambda - a_{p,p}||x_p| = |\sum_{j=1, j\neq p}^{n} a_{p,j}x_j| \leq \sum_{j=1, j\neq p}^{n} |a_{p,j}||x_j|,$$

e quindi dividendo ambo i membri per $|x_p|$

$$|\lambda - a_{p,p}| \leq \sum_{j=1, j\neq p}^{n} |a_{p,j}|\frac{|x_j|}{|x_p|} \leq \sum_{j=1, j\neq p}^{n} |a_{p,j}|.$$

Questa relazione implica che $\lambda \in K_p$ e dunque la tesi. □

Un risultato di inclusione generalmente più debole è fornito dal seguente detto *teorema di Hirsch*.

**Teorema 4.4.2.** Sia $A = (a_{i,j}) \in \mathbb{C}^{n\times n}$ e sia $\| \cdot \|$ una norma matriciale indotta da una norma vettoriale su $\mathbb{C}^n$. Allora

$$\lambda \text{ autovalore di } A \Rightarrow |\lambda| \leq \| A \| .$$

*Dimostrazione.* La relazione $A\boldsymbol{x} = \lambda\boldsymbol{x}$ implica che

$$\| \lambda\boldsymbol{x} \| = |\lambda| \, \| \boldsymbol{x} \| = \| A\boldsymbol{x} \| \leq \| A \| \| \boldsymbol{x} \|$$

da cui la tesi. □

# Lezione 4.5: Esercizi.

**Esercizio 10.** Sia

$$A = \begin{bmatrix} 1 & 1 \\ 1 + 10^{-10} & 1 - 10^{-10} \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

1. Determinare $A^{-1}$.
2. Risolvere il sistema lineare $A\boldsymbol{x} = \boldsymbol{b}$.
3. Posto $\widehat{\boldsymbol{b}} = [1 + \epsilon, 1 - \epsilon]^T$, risolvere il sistema lineare $A\widehat{\boldsymbol{x}} = \widehat{\boldsymbol{b}}$.
4. Verificare che
$$\frac{\| \widehat{\boldsymbol{x}} - \boldsymbol{x} \|}{\| \boldsymbol{x} \|} \leq \| A \| \| A^{-1} \| \frac{\| \boldsymbol{b} - \widehat{\boldsymbol{b}} \|}{\| \boldsymbol{b} \|}$$
per $\| \cdot \| \in \{\| \cdot \|_1, \ \| \cdot \|_2, \ \| \cdot \|_\infty\}$.
5. Mostrare che
$$\| A\widehat{\boldsymbol{x}} - \boldsymbol{b} \|_\infty = |\epsilon|.$$

**Esercizio 11.** Si consideri la matrice triangolare superiore $A = (a_{i,j}) \in \mathbb{R}^{n\times n}$ definita da

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

1. Scrivere una funzione Matlab che dati in input $\boldsymbol{z} \in \mathbb{R}^n$ calcola $\boldsymbol{y} = A \cdot z$ con costo $O(n)$ operazioni aritmetiche.
2. Determinare $|\det(A)|$.
3. Determinare $|\det(D^{-1} \cdot A)|$ dove $D = \text{diag}(\alpha_1, \dots, \alpha_n)$ con
$$\alpha_i = \sqrt{\sum_{k=1}^{n} a_{i,k}^2}, \quad 1 \leq i \leq n.$$

4. Investigare il condizionamento di $A$. In particolare determinare la soluzione del sistema lineare
$$A\boldsymbol{x} = \boldsymbol{e}_n,$$
con $\boldsymbol{e}_n = [0, \dots 0, 1]^T$. Quindi determinare $s_n$ tale che
$$\| A \|_1 \| A^{-1} \|_1 \geq s_n.$$

**Esercizio 12.** Si consideri la matrice tridiagonale
$$A_n(\alpha) = \begin{bmatrix} 2+\alpha & -1 & & \\ -1 & 2+\alpha & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2+\alpha \end{bmatrix} \in \mathbb{R}^{n\times n},\ n \geq 1.$$

1. Dimostrare che $\forall \alpha\ > 0$ $A_n(\alpha)$ è invertibile.
2. Per $\alpha > 0$ sia $\boldsymbol{x} = [x_1, \dots, x_n]$ la soluzione del sistema lineare $A_n(\alpha)\boldsymbol{x} = \boldsymbol{e}_1$, con $\boldsymbol{e}_1 \in \mathbb{R}^n$ la prima colonna della matrice identica di ordine $n$. Dimostrare che $x_n \neq 0$.

**Esercizio 13.** Sia $A = (a_{i,j}) \in \mathbb{R}^{n\times n}$ la matrice tridiagonale con elementi non nulli uguali ad 1 eccetto $a_{n-1,n} = 2$. Per $n = 4$ si ha
$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

1. Si determini $d \neq 0$ tale che posto $D \in \mathbb{R}^{n\times n}$ la matrice diagonale
$$D = \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & & d \end{bmatrix},$$
si ha che $B = D \cdot A \cdot D^{-1}$ è simmetrica.
2. Si dica motivando la risposta se:
   (a) $A$ è simmetrica:
   (b) $A$ ha autovalori reali;
   (c) $A$ è diagonalizzabile.

**Esercizio 14.** Sia $p(z) = \prod_{i=1}^{n}(z - \lambda_i) = \sum_{i=0}^{n-1} a_i z^i + z^n$, $\lambda_i \neq \lambda_j$ se $i \neq j$, un polinomio monico di grado $n$ con zeri distinti e sia $F \in \mathbb{C}^{n\times n}$,
$$F = \begin{bmatrix} 0 & 1 & & & \\ & \ddots & \ddots & & \\ & & 0 & 1 & \\ -a_0 & -a_1 & \dots & \dots & -a_{n-1} \end{bmatrix},$$
la matrice "companion" associata a $p(z)$.

1. Si mostri che per $1 \leq j \leq n$ si ha

$$F\left[1, \lambda_j, \ldots, \lambda_j^{n-1}\right]^T = \lambda_j \left[1, \lambda_j, \ldots, \lambda_j^{n-1}\right]^T.$$

2. Si dimostri che $F$ è diagonalizzabile.

3. Si dimostri che per gli zeri di $p(z)$ vale

$$\max_{1 \leq i \leq n} |\lambda_i| \leq \max\{1, \sum_{i=0}^{n-1} |a_i|\}.$$

**Esercizio 15.** Sia $f\colon \mathbb{R}^n \to \mathbb{R}$, $f(\boldsymbol{v}) = c \parallel \boldsymbol{v} \parallel_1$, $c > 0$.

1. Si mostri che $f$ è una norma vettoriale su $\mathbb{R}^n$.

2. Sia $f\colon \mathbb{R}^{n\times n} \to \mathbb{R}$, $f(A) = c \parallel A \parallel_1$, $c > 0$. Si dica se $f$ è una norma matriciale su $\mathbb{R}^{n\times n}$. In caso di risposta negativa si determinino condizioni su $c$ affinchè $f$ risulti una norma matriciale.

3. Si determini la norma matriciale indotta dalla norma vettoriale definita al punto (1).

**Esercizio 16.** Sia $S \in \mathbb{R}^{n\times n}$ una matrice invertibile, Dimostrare che

$$\parallel \boldsymbol{x} \parallel_S = \parallel S\boldsymbol{x} \parallel_\infty, \quad \boldsymbol{x} \in \mathbb{R}^n,$$

è una norma vettoriale in $\mathbb{R}^n$.

**Esercizio 17.** Siano $A \in \mathbb{R}^{n\times n}$ $L \in \mathbb{R}^{n\times n}$ definite da

$$A = I_n + \gamma L, \quad L = \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & 2 & -1 \\ & & -1 & 2 \end{bmatrix}$$

con $\gamma$ parametro reale.

1. Si determinino i valori del parametro $\gamma$ per cui $A$ risulta predominante diagonale.

2. Per $\gamma = 1$ e $n = 100$ si traccino i cerchi di Gershgorin si dica motivando la risposta se:

   (a) $A$ è invertibile;

   (b) $A$ è definita positiva;

# Capitolo 5

# Metodi Diretti per la Risoluzione di Sistemi Lineari

## Lezione 5.1: Sistemi Triangolari.

Sistemi lineari $A\boldsymbol{x} = \boldsymbol{b}$ dove la matrice dei coefficienti $A \in \mathbb{R}^{n\times n}$ è densa di medie/piccole dimensioni ($n \leq 10^6$) sono generalmente risolti numericamente mediante metodi diretti che con una sequenza finita di trasformazioni elementari riducono il sistema in una forma equivalente facilmente risolubile. Particolare rilevanza in questo contesto assumono le tecniche per la riduzione in forma triangolare.

**Definizione 5.1.1.** Una matrice $T = (t_{i,j}) \in \mathbb{R}^{n\times n}$ si dice *triangolare superiore* se $t_{i,j} = 0$ per $i > j$. Una matrice $T = (t_{i,j}) \in \mathbb{R}^{n\times n}$ si dice *triangolare inferiore* se $t_{i,j} = 0$ per $i < j$.

Si osserva facilmente (si dimostri) che una matrice triangolare $T = (t_{i,j}) \in \mathbb{R}^{n\times n}$ è invertibile se e soltanto se $t_{i,i} \neq 0$ per $1 \leq i \leq n$. Un sistema lineare triangolare $T\boldsymbol{x} = \boldsymbol{b}$, $T = (t_{i,j}) \in \mathbb{R}^{n\times n}$ triangolare superiore (inferiore) invertibile, si risolve con un metodo di *sostituzione all'indietro* (*sostituzione in avanti*) come segue. Dall'ultima equazione si ricava

$$t_{n,n}x_n = b_n \quad \rightarrow \quad x_n = b_n/t_{n,n}.$$

Assumiamo ora di aver determinato $x_{k+1}, \ldots, x_n$ e di voler determinare $x_k$, $1 \leq k \leq n-1$. Dalla $k-$esima equazione si ottiene

$$t_{k,k}x_k + \sum_{j=k+1}^{n} t_{k,j}x_j = b_k \quad \rightarrow \quad x_k = (b_k - \sum_{j=k+1}^{n} t_{k,j}x_j)/t_{k,k}.$$

Il programma seguente implementa il metodo in MatLab.

```
function [x] = solve_tri(t,b)
% backward substitution
```

```
 n=length(b);
 x=zeros(n,1);
 x(n)=b(n)/t(n,n);
 for k=n-1:-1:1
     s=0;
     for j=k+1:n
         s=s+t(k,j)*x(j);
     end
     x(k)=(b(k)-s)/t(k,k);
 end
end
```

Il costo computazionale risulta di $\sum_{k=1}^{n} k = \frac{n(n+1)}{2} = O(n^2)$ operazioni moltiplicative. L'approccio si estende immediatamente a matrici triangolari inferiori.

Se $A \in \mathbb{R}^{n\times n}$ è generale allora per la risoluzione del sistema lineare $A\boldsymbol{x} = \boldsymbol{b}$ si può pensare di ridurre progressivamente $A$ in forma triangolare mediante una sequenza di trasformazioni del tipo

$$A_0 = A, \quad A_k \to A_{k+1},\ 0 \le k \le n-2; \quad \boldsymbol{b}_0 = \boldsymbol{b}, \quad \boldsymbol{b}_k \to \boldsymbol{b}_{k+1}, \quad 0 \le k \le n-2,$$

dove $A_{n-1} = R$ è una matrice triangolare superiore ed i sistemi lineari $A_k\boldsymbol{x} = \boldsymbol{b}_k$, $0 \le k \le n-2$, sono *equivalenti*, i.e., hanno la stessa soluzione. In particolare la soluzione del sistema iniziale $A\boldsymbol{x} = \boldsymbol{b}$ è così ricondotta alla soluzione del sistema finale $A_{n-1}\boldsymbol{x} = R\boldsymbol{x} = \boldsymbol{b}_{n-1}$ in forma triangolare.

Qualora la trasformazione $A_k \to A_{k+1}$ si possa esprimere nella forma $A_{k+1} = E_{k+1}A_k$ con $E_{k+1}$ matrice triangolare inferiore invertibile allora si ottiene

$$R = A_{n-1} = E_{n-1}A_{n-2} = E_{n-1}E_{n-2}A_{n-3} = \ldots = E_{n-1}E_{n-2}\cdots E_1A_0,$$

da cui ponendo $L = (E_{n-1}E_{n-2}\cdots E_1)^{-1} = E_1^{-1}E_2^{-1}\cdots E_{n-1}^{-1}$ segue la fattorizzazione

$$A = A_0 = L \cdot R.$$

**Definizione 5.1.2.** Una matrice $A \in \mathbb{R}^{n\times n}$ si dice *fattorizzabile* nella forma LU se esistono $U \in \mathbb{R}^{n\times n}$ matrice triangolare superiore ed $L \in \mathbb{R}^{n\times n}$ matrice triangolare inferiore con elementi uguali ad 1 sulla diagonale principale tali che $A = L \cdot U$.

Se $A \in \mathbb{R}^{n\times n}$ invertibile è fattorizzata nella forma LU allora dal teorema di Binet segue che $U$ è pure invertibile e dunque il sistema lineare $A\boldsymbol{x} = \boldsymbol{b}$ può essere risolto mediante la sequenza di sistemi triangolari

$$\begin{cases} L\boldsymbol{y} = \boldsymbol{b} \\ U\boldsymbol{x} = \boldsymbol{y} \end{cases}$$

Il seguente risultato fornisce una condizione sufficiente per l'esistenza e l'unicità della fattorizzazione LU di una matrice $A \in \mathbb{R}^{n\times n}$. Seguendo la notazione MatLab indichiamo con $A(1\colon k, 1\colon k) \in \mathbb{R}^{k\times k}$, $1 \le k \le n$, la sottomatrice di $A$ formata dagli elementi situati nelle prime $k$ righe e colonne.

**Teorema 5.1.1.** Sia $A \in \mathbb{R}^{n\times n}$. Se $A(1\colon k, 1\colon k)$ è invertibile per $k = 1, 2, \ldots, n-1$ allora esiste unica la fattorizzazione LU di A.

*Dimostrazione.* La dimostrazione procede per induzione sulla dimensione $n$ della matrice. Per $n = 1$ $A = [a] = [1][a]$ è l'unica fattorizzazione $LU$ di $A$. Supponiamo il teorema vero per matrici di ordine $m \leq n-1$ e dimostriamo per una matrice $A$ di ordine $n$. La relazione $A = LU$ può essere riscritta come

$$\left[\begin{array}{c|c} A(1\colon n-1, 1\colon n-1) & \boldsymbol{z} \\ \hline \boldsymbol{v}^T & \alpha \end{array}\right] = \left[\begin{array}{c|c} L(1\colon n-1, 1\colon n-1) & \boldsymbol{0} \\ \hline \boldsymbol{w}^T & 1 \end{array}\right] \cdot \left[\begin{array}{c|c} U(1\colon n-1, 1\colon n-1) & \boldsymbol{y} \\ \hline \boldsymbol{0}^T & \beta \end{array}\right]$$

dove la matrice $A$ e le matrici incognite $L$ ed $U$ sono partizionate a blocchi con $A(1\colon n-1, 1\colon n-1), L(1\colon n-1, 1\colon n-1), U(1\colon n-1, 1\colon n-1) \in \mathbb{R}^{(n-1)\times(n-1)}$. Questa relazione è equivalente al sistema di equazioni

$$\begin{cases} A(1\colon n-1, 1\colon n-1) = L(1\colon n-1, 1\colon n-1)U(1\colon n-1, 1\colon n-1) \\ \boldsymbol{z} = L(1\colon n-1, 1\colon n-1)\boldsymbol{y} \\ \boldsymbol{v}^T = \boldsymbol{w}^T U(1\colon n-1, 1\colon n-1) \\ \alpha = \boldsymbol{w}^T\boldsymbol{y} + \beta \end{cases}$$

Per ipotesi del teorema le sottomatrici $A(1\colon 1, 1\colon 1), \ldots, A(1\colon n-2, 1\colon n-2)$ della matrice $A(1\colon n-1, 1\colon n-1)$ sono invertibili per cui per l'ipotesi induttiva (teorema vero per matrici di ordine $n-1$) posso concludere l'esistenza e l'unicità della fattorizzazione LU di $A(1\colon n-1, 1\colon n-1)$. Siano pertanto $L(1\colon n-1, 1\colon n-1)$ ed $U(1\colon n-1, 1\colon n-1)$ i fattori triangolari di $A(1\colon n-1, 1\colon n-1)$. Per ipotesi del teorema $A(1\colon n-1, 1\colon n-1)$ è invertibile e quindi $U(1\colon n-1, 1\colon n-1)$ è invertibile e quindi i sistemi lineari che definiscono la seconda e terza equazione ammettono soluzione unica ($L(1\colon n-1, 1\colon n-1)$ è invertibile per definizione). Dati infine $\boldsymbol{w}$ e $\boldsymbol{y}$ l'ultima equazione permette di determinare univocamente il valore di $\beta$. □

Nelle lezioni successive ci porremo il problema del calcolo della fattorizzazione LU e/o della riduzione in forma triangolare di una matrice.

# Lezione 5.2: Matrici Elementari di Gauss ed il Metodo di Eliminazione Gaussiana.

Le matrici elementari di Gauss rappresentano i mattoni per la costruzione di processi di riduzione in forma triangolare e di fattorizzazione triangolare di una matrice.

**Definizione 5.2.1.** Una matrice $E \in \mathbb{R}^{n\times n}$ si dice *elementare di Gauss* se esiste $k \in \mathbb{N}$ con $1 \leq k \leq n$ e $\boldsymbol{v} \in \mathbb{R}^n$ con $v_1 = \ldots v_k = 0$ tale che

$$E = I_n - \boldsymbol{v}\boldsymbol{e}_k^T$$

dove $\boldsymbol{e}_k$ indica il $k-$esimo vettore della base canoninca in $\mathbb{R}^n$.

Si osserva che valgono le seguenti proprietà.

1. Le matrici elementari di Gauss sono matrici triangolari inferiori invertibili con elementi uguali ad 1 sulla diagonale principale.
2. Se $E = I_n - \boldsymbol{v}\boldsymbol{e}_k^T$ è una matrice elementare di Gauss allora $E^{-1} = I_n + \boldsymbol{v}\boldsymbol{e}_k^T$. Infatti vale

$$(I_n - \boldsymbol{v}\boldsymbol{e}_k^T)(I_n + \boldsymbol{v}\boldsymbol{e}_k^T) = I_n + \boldsymbol{v}\boldsymbol{e}_k^T - \boldsymbol{v}\boldsymbol{e}_k^T - \boldsymbol{v}(\boldsymbol{e}_k^T\boldsymbol{v})\boldsymbol{e}_k^T = I_n.$$

3. Sia $\boldsymbol{x} \in \mathbb{R}^n$ con $x_k \neq 0$. Allora esiste una matrice elementare di Gauss $E \in \mathbb{R}^{n\times n}$ tale che $E\boldsymbol{x} = [x_1, \ldots, x_k, 0, \ldots, 0]^T$. Infatti basterà porre $E = I_n - \boldsymbol{v}\boldsymbol{e}_k^T$ con

$$x_j - v_j x_k = 0 \iff v_j = x_j/x_k, \quad k+1 \leq j \leq n.$$

4. Se $E = I_n - \boldsymbol{v}\boldsymbol{e}_k^T$ e $\hat{E} = I_n - \boldsymbol{w}\boldsymbol{e}_\ell^T$ sono matrici elementari di Gauss con $\ell > k$ allora $\boldsymbol{e}_k^T\boldsymbol{w} = 0$ e dunque

$$E \cdot \hat{E} = I_n - \boldsymbol{v}\boldsymbol{e}_k^T - \boldsymbol{w}\boldsymbol{e}_\ell.$$

   Ciò implica che la matrice prodotto risulta costruita semplicemente apponendo nella corretta posizione i vettori $\boldsymbol{v}$ e $\boldsymbol{w}$ dei fattori.

5. Il prodotto $E\boldsymbol{y}$ di una matrice elementare di Gauss $E = I_n - \boldsymbol{v}\boldsymbol{e}_k^T$ per un vettore può essere calcolato con al più $n-k$ operazioni moltiplicative. Si ha infatti che $E\boldsymbol{y} = \boldsymbol{z}$ implica $z_j = y_j$ per $1 \leq j \leq k$ e $z_j = y_j - v_j y_k$ per $j > k$.

Il seguente processo detto *metodo di eliminazione gaussiana* utilizza queste proprietà per la riduzione sotto opportune ipotesi di una matrice $A = A_0$ in forma triangolare superiore. Indichiamo con $\boldsymbol{a}_1^{(k)}, \ldots \boldsymbol{a}_n^{(k)}$ i vettori colonna della matrice $A_k = (a_{i,j}^{(k)})$, $1 \leq i, j \leq n$, $0 \leq k \leq n-1$. Se assumiamo che $a_{1,1}^{(0)} \neq 0$ allora per la proprietà (3) possiamo determinare $E_1$ tale da aversi

$$E_1\boldsymbol{a}_1^{(0)} = \left[a_{1,1}^{(0)}, 0, \ldots, 0\right]^T.$$

Risulta

$$E_1 = I_n - \left[0, a_{2,1}^{(0)}/a_{1,1}^{(0)}, \ldots, a_{n,1}^{(0)}/a_{1,1}^{(0)}\right]^T \boldsymbol{e}_1^T.$$

I termini $m_{2,1}^{(0)} = a_{2,1}^{(0)}/a_{1,1}^{(0)}, \ldots, m_{n,1}^{(0)} = a_{n,1}^{(0)}/a_{1,1}^{(0)}$ sono detti *moltiplicatori di Gauss* mentre il termine $a_{1,1}^{(0)}$ è detto *pivot* o *elemento pivotale*. Poniamo dunque

$$A_1 = E_1 A_0, \quad \boldsymbol{b}_1 = E_1\boldsymbol{b}_0.$$

Il processo prosegue operando sulla matrice $A_1$. Se assumiamo che $a_{2,2}^{(1)} \neq 0$ allora per la proprietà (3) possiamo determinare $E_2$ tale da aversi

$$E_2\boldsymbol{a}_2^{(1)} = \left[a_{1,2}^{(1)}, a_{2,2}^{(1)}, 0, \ldots, 0\right]^T.$$

Risulta

$$E_2 = I_n - \left[0, 0, a_{3,2}^{(1)}/a_{2,2}^{(1)}, \ldots, a_{n,2}^{(1)}/a_{2,2}^{(0)}\right]^T \boldsymbol{e}_2^T.$$

Si osserva che $E_2\boldsymbol{a}_1^{(1)} = a_1^{(1)}$. Poniamo dunque

$$A_2 = E_2 A_1, \quad \boldsymbol{b}_2 = E_2\boldsymbol{b}_1.$$

In questo modo assumendo che valga

$$a_{j,j}^{(j-1)} \neq 0, \quad 1 \leq j \leq n-1, \tag{5.1}$$

è possibile determinare una sequenza di matrici elementari di Gauss $E_1, \dots E_{n-1}$ tali da aversi

$$E_{n-1}E_{n-2}\cdots E_1 A_0 = E_{n-1}E_{n-2}\cdots E_1 A = A_{n-1} = R,$$

con $R = A_{n-1}$ matrice triangolare superiore. Le relazioni $A_k = E_k A_{k-1}$, $\boldsymbol{b}_k = E_k \boldsymbol{b}_{k-1}$ espresse in termini di componenti si scrivono

$$\begin{cases} a_{i,j}^{(k)} = a_{i,j}^{(k-1)} \text{ se } i \le k \text{ o } j \le k-1; \\ a_{i,k}^{(k)} = 0 \text{ se } i > k; \\ a_{i,j}^{(k)} = a_{i,j}^{(k-1)} - m_{i,k}^{(k-1)} a_{k,j}^{(k-1)} \text{ se } i > k \text{ e } j > k; \end{cases} \tag{5.2}$$

$$\begin{cases} b_i^{(k)} = b_i^{(k-1)} \text{ se } i \le k; \\ b_i^{(k)} = b_i^{(k-1)} - m_{i,k}^{(k-1)} b_k^{(k-1)} \text{ se } i > k. \end{cases} \tag{5.3}$$

La risoluzione del sistema lineare $A\boldsymbol{x} = \boldsymbol{b}$ viene dunque ricondotta alla risoluzione del sistema triangolare

$$R\boldsymbol{x} = \boldsymbol{b}_{n-1} = E_{n-1}E_{n-2}\cdots E_1 \boldsymbol{b}.$$

Inoltre si ha che

$$A = E_1^{-1}E_2^{-1}\cdots E_{n-1}^{-1}R.$$

Dalla proprietà (2) segue che le inverse delle matrici elementari sono matrici elementari ottenute semplicemente cambiando il segno del vettore $\boldsymbol{v}$. Dalla proprietà (4) segue che il prodotto delle matrici elementari $L = E_1^{-1}E_2^{-1}\cdots E_{n-1}^{-1}$ è determinato apponendo nel corretto ordine i moltiplicatori di Gauss, i.e.,

$$L = \begin{bmatrix} 1 & & & & \\ \frac{a_{2,1}^{(0)}}{a_{1,1}^{(0)}} & 1 & & & \\ \frac{a_{3,1}^{(0)}}{a_{1,1}^{(0)}} & \frac{a_{3,2}^{(1)}}{a_{2,2}^{(1)}} & \ddots & & \\ \vdots & \vdots & \vdots & \ddots & \\ \frac{a_{n,1}^{(0)}}{a_{1,1}^{(0)}} & \frac{a_{n,2}^{(1)}}{a_{2,2}^{(1)}} & \cdots & \frac{a_{n,n-1}^{(n-2)}}{a_{n-1,n-1}^{(n-2)}} & 1 \end{bmatrix}$$

Per (5.2) e (5.3) si ha che se applicabile il metodo di eliminazione gaussiana permette la risoluzione del sistema lineare $A\boldsymbol{x} = \boldsymbol{b}$ e il calcolo della fattorizzazione LU di $A$ con $\sum_{k=1}^{n-1} k^2 + 2\sum_{k=1}^{n-1} k = n^3/3 + O(n^2)$ operazioni moltiplicative. Il seguente programma MatLab realizza il calcolo della fattorizzazione LU di $A$ e la risoluzione del sistema lineare $A\boldsymbol{x} = \boldsymbol{b}$.

```
function [a,x] = gauss(a,b)
n=length(b);
m=zeros(n,1);
for k=1:n-1
    % calcolo dei moltiplicatori
    % che vengono sovrascritti in a
    for i=k+1:n
        m(i)=a(i,k)/a(k,k);
```

```
        a(i,k)=m(i);
    end
    %calcolo della trasformazione
    for i=k+1:n
        for j=k+1:n
            a(i,j)=a(i,j)-m(i)*a(k,j);
        end
        b(i)=b(i)-m(i)*b(k);
    end
    % risoluzione del sistema triangolare
  x=solve_tri(triu(a), b);
end
```

Concludiamo questa lezione osservando che la condizione sufficiente per l'esistenza e l'unicità della fattorizzazione LU stabilita nel teorema (5.1.1) coincide con la condizione (5.1) per l'applicabilità del metodo di Gauss. Infatti vale il seguente risultato.

**Teorema 5.2.1.** Sia $A \in \mathbb{R}^{n\times n}$. Allora $A(1\colon k, 1\colon k)$ è invertibile per $k = 1, 2, \ldots, j \leq n-1$ se e soltanto se $a_{k,k}^{(k-1)} \neq 0$ per $k = 1, 2, \ldots, j \leq n-1$.

*Dimostrazione.* Si dimostra per indizione su $j$. Per $j = 1$ segue immediatamente da $A(1\colon 1, 1\colon 1) = a_{1,1}^{(0)}$. Assumiamo il risultato vero per $k \leq j-1$ e dimostriamo per $k = j$. Dalla relazione

$$E_{j-1}E_{j-2}\cdots E_1 A = F_j A = A_{j-1}$$

segue che $F_j$ è triangolare inferiore con elementi uguali ad 1 sulla diagonale principale ed inoltre

$$F_j(1\colon \ell, 1\colon \ell)A(1\colon \ell, 1\colon \ell) = \begin{bmatrix} a_{1,1}^{(0)} & \cdots & \cdots & \cdots \\ 0 & a_{2,2}^{(1)} & \cdots & \cdots \\ & & \ddots & \vdots \\ & & & a_{\ell,\ell}^{(\ell-1)} \end{bmatrix}, \quad 1 \leq \ell \leq j.$$

Si ha dunque che $a_{k,k}^{(k-1)} \neq 0$ per $k = 1, 2, \ldots, j$ se e soltanto se $a_{k,k}^{(k-1)} \neq 0$ per $k = 1, 2, \ldots, j-1$ e $a_{j,j}^{(j-1)} \neq 0$ e quindi per ipotesi induttiva se e soltanto se $A(1\colon k, 1\colon k)$ è invertibile per $k = 1, 2, \ldots, j-1$ e $a_{j,j}^{(j-1)} = \dfrac{\det(A(1\colon j, 1\colon j))}{\det(A(1\colon j-1, 1\colon j-1))} \neq 0$ e dunque se e soltanto se $A(1\colon k, 1\colon k)$ è invertibile per $k = 1, 2, \ldots, j$. □

# Lezione 5.3: Il Metodo di Gauss per Matrici Invertibili: Tecniche di Pivoting e Stabilità.

L'estensione del metodo di eliminazione gaussiana ad una generica matrice invertibile $A \in \mathbb{R}^{n\times n}$ senza assunzioni sull'invertibilità delle sue sottomatrici avviene mediante l'introduzione di opportune tecniche di riordinamento delle equazioni e/o delle variabili. Si osservi che dall'invertibilità di $A = A_0$ segue che esiste un

elemento non nullo nella prima colonna, i.e., $\exists j\colon a_{j,1}^{(0)} \neq 0$. Detta $P_1$ allora la matrice di permutazione ottenuta dalla matrice $I_n$ scambiando tra loro la prima e la $j-$ esima colonna si ha che $(P_1A_0)_{1,1} = a_{j,1}^{(0)}$ e dunque posso determinare $E_1$ tale da aversi $E_1P_1A_0\boldsymbol{e}_1 = \left[a_{j,1}^{(0)}, 0, \ldots, 0\right]^T$. Poniamo

$$A_1 = E_1P_1A_0, \quad \boldsymbol{b}_1 = E_1P_1\boldsymbol{b}_0.$$

Dall'invertibilità di $A = A_0$ segue l'invertibilità di $A_1$ ed inoltre dalla regola di Laplace per il calcolo del determinante si ha

$$\det(A_1) = a_{j,1}^{(0)}\det(A_1(2\colon n, 2\colon n)),$$

per cui esiste un elemento non nullo nella prima colonna di $A_1(2\colon n, 2\colon n)$. Detti $a_{\ell,2}^{(1)}$, $\ell \geq 2$, questo elemento, $P_2$ la matrice di permutazione ottenuta dalla matrice $I_n$ scambiando tra loro la seconda e la $\ell-$ esima colonna e $E_2$ la matrice elementare di Gauss tale che $E_2P_2A_1\boldsymbol{e}_2 = \left[a_{1,2}^{(1)}, a_{\ell,2}^{(1)}, 0, \ldots, 0\right]^T$ si ha

$$A_2 = E_2P_2A_1, \quad \boldsymbol{b}_2 = E_2P_2\boldsymbol{b}_1.$$

Il processo così procede determinando matrici elementari gi Gauss $E_1, \ldots E_{n-1}$ e matrici di permutazione (scambio) $P_1, \ldots P_{n-1}$ tale che

$$E_{n-1}P_{n-1}E_{n-2}P_{n-2}\cdots E_1P_1A = R, \quad E_{n-1}P_{n-1}E_{n-2}P_{n-2}\cdots E_1P_1\boldsymbol{b} = \boldsymbol{b}_{n-1}.$$

La risoluzione del sistema lineare $A\boldsymbol{x} = \boldsymbol{b}$ è ricondotta alla risoluzione del sistema lineare $R\boldsymbol{x} = \boldsymbol{b}_{n-1}$. Inoltre posto

$$L = (E_{n-1}P_{n-1}E_{n-2}P_{n-2}\cdots E_1P_1)^{-1} = P_1^TE_1^{-1}\cdots P_{n-2}^TE_{n-2}^{-1}P_{n-1}^TE_{n-1}^{-1},$$

si può ancora scrivere

$$A = L \cdot R$$

ma la matrice $L$ non risulta generalmente triangolare inferiore (MatLab definisce $L$ " "psychologically lower triangular matrix" (i.e. a product of lower triangular and permutation matrices")). Si osserva comunque che

$$P_2E_1 = P_2(I_n - \boldsymbol{v}\boldsymbol{e}_1^T) = (I_n - \hat{\boldsymbol{v}}\boldsymbol{e}_1^T)P_2,$$

per cui utlizzando ripetutamente questa proprietà di sostanziale commutatività si può scrivere

$$E_{n-1}P_{n-1}E_{n-2}P_{n-2}\cdots E_1P_1 = (\hat{E}_{n-1}\hat{E}_{n-2}\cdots\hat{E}_1)(P_{n-1}P_{n-2}\cdots P_1),$$

da cui ponendo

$$\hat{L} = (\hat{E}_{n-1}\hat{E}_{n-2}\cdots\hat{E}_1)^{-1}, \quad P = P_{n-1}P_{n-2}\cdots P_1,$$

si perviene alla conclusione che il metodo di eliminazione con scambi di righe calcola la fattorizzazione LU di una matrice permutata, i.e.,

$$PA = \hat{L}U.$$

La scelta dell'elemento pivotale è suggerita da valutazioni di stabilità numerica. Si può infatti dimostrare un risultato di stabilità all'indietro per cui se indichiamo con $\tilde{L}$ il fattore $L$ effettivamente calcolato ed analogamente $\tilde{R}$ il fattore $R$ effettivamente calcolato allora

$$\tilde{L}\tilde{R} = A + E, \quad \frac{\| E \|}{\| L \| \| R \|} = O(u). \tag{5.4}$$

Per minimizzare la norma della perturbazione è dunque essenziale evitare la crescita dei moduli degli elementi in $L$ ed $U$. Per controllare gli elementi di $L$ si può scegliere come pivot l'elemento di modulo massimo sulla colonna corrente, i.e. se $|a_{j,k}^{(k-1)}| = \max_{k \le i \le n} |a_{i,k}^{(k-1)}|$ allora si scambia la riga $k$ con la riga $j$ al passo $k$. Questa tecnica detta del *massimo pivot parziale* garantisce che gli elementi di $L$ hanno modulo minore o uguale ad 1. Gli elementi di $U$ possono comunque crescere ma generalmente ciò non accade ed il metodo risultante è suggerito come metodo di scelta per la risoluzione di sistemi lineari densi di medie/piccole dimensioni (operatore "backslash" in MatLab). Il seguente programma MatLab implementa il metodo di eliminazione gaussiana per la risoluzione del sistema lineare $A\boldsymbol{x} = \boldsymbol{b}$.

```
function [x]= gauss_pp(a,b)
%IL programma non esegue esplicitamente lo scambio tra le righe
% ma tiene traccia nel vettore nriga
n=length(b);
nriga=zeros(n,1);
x=zeros(n,1);
m=zeros(n,1);
for i=1:n
    nriga(i)=i;
end
%%riduzione in forma triangolare
for k=1:n-1;
max=0;
index=0;
%trova il pivot  nella colonna corrente
for j=k:n
    if abs(a(nriga(j),k))>max
        max=abs(a(nriga(j),k));
        index=j;
    end
end
%aggiorna il vettore nriga
if nriga(k)~=nriga(index)
    nn=nriga(k);
    nriga(k)=nriga(index);
    nriga(index)=nn;
end
%passo di eliminazione
for i=k+1:n
    m(nriga(i))=a(nriga(i),k)/a(nriga(k),k);
    a(nriga(i),k)=0;
```

```
    for j=k+1:n
        a(nriga(i),j)=a(nriga(i),j)-m(nriga(i))*a(nriga(k),j);
    end
    b(nriga(i))=b(nriga(i))-m(nriga(i))*b(nriga(k));
end
end
%risoluzione del sistema triangolare
x=solve_tri(triu(a(nriga,:)), b(nriga));
end
```

Il lettore confronti sperimentalmente i risultati generati dal programma con i risultati forniti dal risolutore (backslash) di MatLab.

## Lezione 5.4: Esercizi.

**Esercizio 18.** Si considera il modello aperto di Leontief per descrivere l'economia di uno stato. Denotiamo con $S_1, \dots S_n$ le industrie, con $x_j$, $1 \leq j \leq n$, la quantità totale del bene prodotto dall'industria $S_j$ e con $a_{i,j}x_j$ la quantità del bene $x_i$ consumata dall'industria $S_j$ per la produzione del bene $x_j$. Denotiamo infine con $b_j$, $1 \leq j \leq n$, la domanda esterna del bene $x_j$. Assumendo che per ogni bene la produzione totale eguaglia il consumo totale (incluso la domanda esterna) si ottiene il seguente sistema di equazioni lineari nelle incognite $x_1, \dots, x_n$:

$$\begin{cases} x_1 = \sum_{j=1}^{n} a_{1,j}x_j + b_1 \\ x_2 = \sum_{j=1}^{n} a_{2,j}x_j + b_2 \\ \vdots \qquad \vdots \\ x_n = \sum_{j=1}^{n} a_{n,j}x_j + b_n \end{cases} \tag{5.5}$$

1. Si descriva la matrice dei coefficienti $A \in \mathbb{R}^{n\times n}$ del sistema lineare (5.5).

2. Posto $a_{i,j} = a > 0$, $1 \leq i, j \leq n$, si determinino condizioni sufficienti a garantire l'esistenza e l'unicità della fattorizzazione LU di $A$.

3. Posto $a_{i,j} = a > 0$, $1 \leq i, j \leq n$, si dimostri che $A$ è una matrice elementare e se ne fornisca la rappresentazione.

4. Posto $a_{i,j} = a > 0$, $1 \leq i, j \leq n$, si indaghi sperimentalmente l'invertibilità di $A$. In particolare calcolare l'inversa e verificare sperimentalmente se $A^{-1}$ (quando esiste) è una matrice elementare. In caso affermativo determinarne una rappresentazione e dimostrare teoricamente i risultati ottenuti.

**Esercizio 19.** Si consideri la matrice $A = (a_{i,j}) \in \mathbb{R}^{n\times n}$ definita da

$$A_n = \begin{bmatrix} 1 & -1 & \dots & \dots & -1 \\ & 1 & -1 & \dots & -1 \\ & & \ddots & \ddots & \vdots \\ & & & 1 & -1 \\ 1 & & & & 1 \end{bmatrix}.$$

Per $n = 4$ si ha

$$A_4 = \begin{bmatrix} 1 & -1 & -1 & -1 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & 1 & -1 \\ 1 & 0 & 0 & 1 \end{bmatrix}.$$

1. Si dica se la matrice $A_n$, ammette fattorizzazione LU.
2. In caso affermativo si determini la fattorizzazione LU.
3. Scrivere una funzione MatLab che dati in input $n$ implementa il metodo di Gauss e restituisce in output il fattore $U$.
4. Scrivere una funzione MatLab che dati in input $n$ e $\boldsymbol{b}$ costruisce $U$ e risolve il sistema lineare $U\boldsymbol{x} = \boldsymbol{b}$. Valutarne il costo computazionale.

**Esercizio 20.** Si consideri il sistema lineare nelle incognite $x_1, \ldots, x_n$,

$$\begin{cases} x_1 - \sum_{j=2}^{n} x_j = b_1 \\ x_2 - \sum_{j=3}^{n} x_j = b_2 \\ \vdots \\ x_{n-1} - x_n = b_{n-1} \\ x_n = b_n \end{cases}$$

1. Si determini la matrice $A$ dei coefficienti del sistema.
2. Si determini la matrice $A^{-1}$.
3. Si determini il numero di condizionamento del sistema in norma infinito.
4. Si scriva un programma MatLab che risolve il sistema con costo lineare senza memorizzare esplicitamente la matrice.

**Esercizio 21.** La seguente matrice $A_n \in \mathbb{R}^{n\times n}$, $n \geq 3$, si incontra nella teoria dell'approssimazione mediante funzioni spline

$$A_n = \begin{bmatrix} 2(\gamma_1+\gamma_n) & \gamma_1 & 0 & \ldots & 0 & \gamma_n \\ \gamma_1 & 2(\gamma_1+\gamma_2) & \gamma_2 & \ddots & & 0 \\ 0 & \gamma_2 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & & \ddots & \ddots & \ddots & \gamma_{n-1} \\ \gamma_n & 0 & \ldots & 0 & \gamma_{n-1} & 2(\gamma_{n-1}+\gamma_n) \end{bmatrix},$$

con $\gamma_1, \ldots, \gamma_n > 0$. Ad esempio per $n = 5$ si ha

$$A_5 = \begin{bmatrix} 2(\gamma_1+\gamma_5) & \gamma_1 & 0 & 0 & \gamma_5 \\ \gamma_1 & 2(\gamma_1+\gamma_2) & \gamma_2 & 0 & 0 \\ 0 & \gamma_2 & 2(\gamma_2+\gamma_3) & \gamma_3 & 0 \\ 0 & 0 & \gamma_3 & 2(\gamma_3+\gamma_4) & \gamma_4 \\ \gamma_5 & 0 & 0 & \gamma_4 & 2(\gamma_4+\gamma_5) \end{bmatrix},$$

con $\gamma_1, \ldots, \gamma_5 > 0$.

1. Dimostrare che $A_n$ è invertibile.
2. Dimostrare che $A_n$ ammette fattorrizzazione LU.
3. Scrivere una funzione MatLab che dati in input $n$ e $\gamma_1, \ldots, \gamma_n$ restituisce la fattorizzazione LU di $A_n$. Valutarne il costo computazionale.

**Esercizio 22.** Si consideri la matrice $A = (a_{i,j}) \in \mathbb{R}^{n\times n}$, $n \geq 2$, definita da

$$a_{i,j} = \begin{cases} 1 \text{ se } i = j; \\ -1 \text{ se } i = j-1; \\ \rho \text{ se } i = n \text{ e } j = 1; \\ 0 \text{ altrimenti.} \end{cases}$$

1. Dimostrare che $A$ ammette fattorizzazione LU $\forall \rho \in \mathbb{R}$.
2. Dimostrare che per il fattore triangolare superiore $U = (u_{i,j}) \in \mathbb{R}^{n\times n}$ si ha
$$u_{i,j} = \begin{cases} 1 \text{ se } i = j \text{ e } i < n; \\ 1+\rho \text{ se } i = j = n; \\ -1 \text{ se } i = j-1; \\ 0 \text{ altrimenti.} \end{cases}$$
3. Determinare per quali valori del parametro $\rho \in \mathbb{R}$ la fattorizzazione LU è unica.
4. Scrivere una funzione MatLab che dati in input $n \in \mathbb{N}$, $n \geq 2$, $\rho \in \mathbb{R}$ implementa un processo di sostituzione all'indietro e restituisce in output il vettore $\boldsymbol{z}$ soluzione del sistema lineare
$$U\boldsymbol{z} = \texttt{ones}(n, 1).$$
5. Valutare il costo aritmetico dell'algoritmo.
6. Per $n = 200, 400, 800$ e $\rho = -0.99$ riportare l'errore relativo
$$\epsilon_n = \frac{\| \boldsymbol{x} - \boldsymbol{z} \|_1}{\| \boldsymbol{x} \|_1},$$
ove $\boldsymbol{x} = [n-1 : -1 : 0]^T + \dfrac{1}{1+\rho} * \texttt{ones}(n, 1)$ e $\boldsymbol{z}$ è la soluzione del sistema ottenuta con la procedura implementata al punto precedente.

**Esercizio 23.** Sia $\boldsymbol{x} = [x_1, \ldots, x_{n-1}]^T \in \mathbb{R}^{n-1}$ e definiamo

$$\mathcal{A}(\boldsymbol{x}) = \begin{bmatrix} 1 & & & -x_1 \\ & \ddots & & \vdots \\ & & 1 & -x_{n-1} \\ x_1 & \ldots & x_{n-1} & 1 \end{bmatrix} \in \mathbb{R}^{n\times n}$$

la matrice "a freccia" generata da $\boldsymbol{x}$.

1. Si dimostri che esiste ed è unica la fattorizzazione LU di $\mathcal{A}(\boldsymbol{x})$.

2. Si dimostri che il fattore triangolare superiore è

$$\mathcal{U}(\boldsymbol{x}) = \begin{bmatrix} 1 & & & -x_1 \\ & \ddots & & \vdots \\ & & 1 & -x_{n-1} \\ & & & 1+ \| \boldsymbol{x} \|_2^2 \end{bmatrix}.$$

3. Scrivere una funzione MatLab che dati in input $\boldsymbol{x} \in \mathbb{R}^{n-1}$ e $\boldsymbol{b} \in \mathbb{R}^n$ implementa un processo di sostituzione all'indietro e restituisce in output $\boldsymbol{z} \in \mathbb{R}^n$ soluzione del sistema lineare

$$\mathcal{U}(\boldsymbol{x}) \cdot \boldsymbol{z} = \boldsymbol{b}.$$

4. Valutare il costo aritmetico dell'algoritmo.

5. Posto $n = 100$, $\boldsymbol{x}_\theta = \theta \boldsymbol{e}$, $\boldsymbol{e} = [1, 1/2, \ldots, 1/99]^T$, $\boldsymbol{b}_\theta = \mathcal{U}(\boldsymbol{x}_\theta) \cdot [\boldsymbol{e}^T, 1]^T$ e $\boldsymbol{z}_\theta$ la soluzione del sistema lineare

$$\mathcal{U}(\boldsymbol{x}_\theta) \cdot \boldsymbol{z}_\theta = \boldsymbol{b}_\theta.$$

restituita dall'algoritmo, si riportino gli errori

$$\epsilon_\theta = \frac{\| \boldsymbol{z}_\theta - [\boldsymbol{e}^T, 1]^T \|_2}{\| [\boldsymbol{e}^T, 1]^T \|_2}$$

per $\theta = 10^2, 10^4, 10^6, 10^8$.

**Esercizio 24.** Sia $A_n \in \mathbb{R}^{n \times n} = (a_{i,j})$, $n \geq 3$, definita da

$$a_{i,j} = \begin{cases} 1 \text{ se } i = j \text{ o } i = j - 1: \\ -1 \text{ se } j = 1, 2 \text{ e } i = j + 1, \ldots, n; \\ 0 \text{ altrimenti} \end{cases}$$

Per $n = 4$ si ha

$$A_4 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ -1 & 1 & 1 & 0 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 0 & 1 \end{bmatrix}.$$

1. Si determini la matrice elementare di Gauss $E_1$ tale che

$$E_1 \left[a_{1,1}, \ldots, a_{n,1}\right]^T = \left[a_{1,1}, 0, \ldots, 0\right]^T.$$

2. Si mostri che $B = E_1 \cdot A_n$ risulta triangolare superiore. Si determini quindi una fattorizzazione triangolare di $A_n$.

3. Si dica se tale fattorizzazione è unica.

4. Scrivere una funzione MatLab che dato in input $n \in \mathbb{N}$ e $\boldsymbol{b} \in \mathbb{R}^n$ restituisce in output la soluzione $\boldsymbol{x}$ del sistema lineare $B\boldsymbol{x} = E_1 \boldsymbol{b}$.

5. Determinare il costo computazionale dell'algoritmo.

6. Per $n \in \{64, 128, 256\}$ e $\boldsymbol{b} = \texttt{ones}(n, 1)$ riportare l'errore relativo $\epsilon_n$,

$$\epsilon_n = \frac{\| \boldsymbol{x} - \hat{\boldsymbol{x}} \|_1}{\| \hat{\boldsymbol{x}} \|_1},$$

tra la soluzione $\boldsymbol{x}$ calcolata dall'algoritmo e la soluzione del sistema lineare $A_n \hat{\boldsymbol{x}} = \boldsymbol{b}$ calcolata mediante l'operatore "backslash" di Matlab.

**Esercizio 25.** Sia $W \in \mathbb{R}^{n \times 2}$, $n \geq 2$. Denotiamo con $\boldsymbol{w}_1$, $\boldsymbol{w}_2$ rispettivamente la prima e seconda colonna di $W$ e sia $\boldsymbol{w}_1 \neq \boldsymbol{0}$.

1. Si dimostri che esiste ed è unica la fattorizzazione LU di $A = W^T \cdot W$. Determinare tale fattorizzazione e siano $\hat{L}$ e $\hat{U}$ i fattori triangolari.

2. Posto $B \in \mathbb{R}^{(n+2)\times(n+2)}$ definita da

$$B = \left[\begin{array}{c|c} I_n & W \\ \hline W^T & 0_2 \end{array}\right],$$

con $0_2$ matrice nulla di ordine 2, si mostri che $B$ ammette fattorizzazione LU con fattori triangolari definiti da

$$L = \left[\begin{array}{c|c} I_n & \\ \hline W^T & \hat{L} \end{array}\right], \quad U = \left[\begin{array}{c|c} I_n & W \\ \hline & -\hat{U} \end{array}\right].$$

Si dica motivando la risposta se tale fattorizzazione è unica.

3. Scrivere una funzione MatLab che dati in input $\boldsymbol{b} \in \mathbb{R}^{n+2}$, $\boldsymbol{w}_1 \in \mathbb{R}^n$ e $\boldsymbol{w}_2 \in \mathbb{R}^n$ senza memorizzare esplicitamente la matrice $U$ implementa un metodo di sostituzione all'indietro restituendo in output il vettore $\boldsymbol{x}$ soluzione del sistema lineare $U\boldsymbol{x} = \boldsymbol{b}$.

4. Valutare il costo computazionale dell'algoritmo.

5. Per $n = 64$, $\boldsymbol{b} = \boldsymbol{e}_{n+2}$, $\boldsymbol{w}_1 = \texttt{ones}(n, 1)$, $\boldsymbol{w}_2 = -\boldsymbol{e}_n$ riportare il valore di $\| B\boldsymbol{x} - \boldsymbol{b} \|_2$ con $\boldsymbol{x}$ vettore soluzione restituito dall'algoritmo.

**Esercizio 26.** Sia $A_\alpha \in \mathbb{R}^{n \times n}$, $n > 2$, la matrice definita da

$$A_\alpha = I_n - \alpha \boldsymbol{e}\boldsymbol{e}^T,$$

con $I_n$ matrice identità di ordine $n$ e $\boldsymbol{e} = [1, \ldots, 1]^T \in \mathbb{R}^n$.

1. Determinare l'insieme $\mathcal{S}$ dei valori del parametro reale $\alpha$ per cui $A_\alpha$ risulta predominante diagonale. Si dica inoltre se le seguenti proposizioni sono vere o false:

   (a) $\alpha \in \mathcal{S} \ \rightarrow \ A_\alpha$ è invertibile;

   (b) $\alpha \notin \mathcal{S} \rightarrow \ A_\alpha$ non è invertibile.

2. Per $\alpha = 2/n$ si mostri che

$$A_\alpha^{-1} = A_\alpha.$$

3. Per $\alpha = 2/n$ si determini il numero di condizionamento $\mathcal{K}_\infty(A_\alpha)$ di $A_\alpha$ in norma infinito mostrando che vale

$$\mathcal{K}_\infty(A_\alpha) \leq 9, \quad \forall n > 2.$$

4. Scrivere una funzione MatLab che dati in input $n \in \mathbb{N}$ e $\boldsymbol{b} \in \mathbb{R}^n$ risolve il sistema lineare $A_\alpha \boldsymbol{x} = \boldsymbol{b}$ per $\alpha = 2/n$ restituendo in uscita il vettore $\boldsymbol{x} = A_\alpha \boldsymbol{b}$. L'implementazione non deve richiedere la memorizzazione esplita della matrice e l'algoritmo per il calcolo del prodotto matrice vettore deve avere costo lineare nella dimensione.

5. Osservato che per $\alpha = 2/n$ si ha $A_\alpha \boldsymbol{e} = -\boldsymbol{e}$, per $n = 49k, k \in \{1, 2, 4\}$ e $\boldsymbol{b} = -\boldsymbol{e}$ riportare gli errori

$$\epsilon_n = \| x - e \|_\infty, \quad n = 49k,\ k = 1, 2, 4,$$

   dove $x$ è l'approssimazione della soluzione restituita dal programma.

6. Si dica se vale $\epsilon_n \leq 9$ eps, $n = 49k$, $k = 1, 2, 4$, con eps precisione di macchina.

**Esercizio 27.** Si consideri la matrice tridiagonale $A_n \in \mathbb{R}^{n \times n}$ definita da

$$A = \begin{bmatrix} 0 & 2 & & \\ 1 & 0 & \ddots & \\ & \ddots & \ddots & 2 \\ & & 1 & 0 \end{bmatrix}.$$

1. Sia $D = \mathrm{diag}(1, 2, 4, \ldots, 2^{n-1})$. Si determini $B_n = D^{1/2} \cdot A_n \cdot D^{-1/2}$.

2. Si dimostri che $A_n$ ha autovalori reali.

3. Per $n \geq 2$ si dica se $A_n$ ammette fattorizzazione LU.

4. Si dimostri che $A_n + 4I_n$ è invertibile per ogni $n \geq 1$.

5. Determinare un insieme di inclusione per gli autovalori di $A_n + 4I_n$, $n \geq 1$.

6. Analizzare sperimentalmente il costo computazionale della risoluzione di un sistema lineare $(A_n + 4I_n)\boldsymbol{x}_n = \boldsymbol{b}_n$ dove $\boldsymbol{b}_n$ è un vettore generato casualmente mediante la funzione `rand` ed il sistema lineare è risolto mediante la routine `tridsolve.m` riportata di seguito.

```
function x = tridisolve(a,b,c,d)
%   TRIDISOLVE  Solve tridiagonal system of equations.
%     x = TRIDISOLVE(a,b,c,d) solves the system of linear equations
%     b(1)*x(1) + c(1)*x(2) = d(1),
%     a(j-1)*x(j-1) + b(j)*x(j) + c(j)*x(j+1) = d(j), j = 2:n-1,
%     a(n-1)*x(n-1) + b(n)*x(n) = d(n).
%
%   The algorithm does not use pivoting, so the results might
%   be inaccurate if abs(b) is much smaller than abs(a)+abs(c).
%   More robust, but slower, alternatives with pivoting are:
%     x = T\d where T = diag(a,-1) + diag(b,0) + diag(c,1)
%     x = S\d where S = spdiags([[a; 0] b [0; c]],[-1 0 1],n,n)

%   Copyright 2014 Cleve Moler
```

```
%   Copyright 2014 The MathWorks, Inc.

x = d;
n = length(x);

for j = 1:n-1
   mu = a(j)/b(j);
   b(j+1) = b(j+1) - mu*c(j);
   x(j+1) = x(j+1) - mu*x(j);
end

x(n) = x(n)/b(n);
for j = n-1:-1:1
   x(j) = (x(j)-c(j)*x(j+1))/b(j);
end
```

**Esercizio 28.** Sia $A = (a_{i,j}) \in \mathbb{R}^{n\times n}$ definita da

$$a_{i,j} = \begin{cases} 1 \text{ se } i \geq j \text{ o } j = n; \\ 0 \text{ altrimenti.} \end{cases}$$

1. Si mostri che $A$ ammette fattorizzazione LU.
2. Si determinino i fattori triangolari $L$ ed $U$ di tale fattorizzazione.
3. Si determini $\det(A)$.
4. Si caratterizzi l'insieme $Im(A) = \{\boldsymbol{z} \in \mathbb{R}^n \colon \exists \boldsymbol{x}, A\boldsymbol{x} = \boldsymbol{z}\}$.
5. Si determini il costo computazionale della risoluzione del sistema lineare $A\boldsymbol{x} = \boldsymbol{b}$, $b \in Im(A)$, mediante la procedura
   (a) $L\boldsymbol{y} = \boldsymbol{b}$;
   (b) $U\boldsymbol{x} = \boldsymbol{y}$.
6. Si scriva un programma MatLab che implementa la procedura descritta al punto precedente.

# Capitolo 6

# Metodi Iterativi per la Risoluzione di Sistemi Lineari

## Lezione 6.1: Generalità sui Metodi Iterativi.

Sistemi lineari $A\boldsymbol{x} = \boldsymbol{b}$ dove la matrice dei coefficienti $A \in \mathbb{R}^{n\times n}$ è sparsa o di elevate dimensioni ($n > 10^6$) sono generalmente risolti numericamente mediante metodi iterativi che a partire da un vettore iniziale $\boldsymbol{x}^{(0)}$ generano una sequenza di approssimazioni $\boldsymbol{x}^{(k)}$, $k > 0$, che converge alla soluzione del sistema lineare, i.e.,

$$\lim_{k\to+\infty} \| \boldsymbol{x}^{(k)} - \boldsymbol{x} \| = 0.$$

In pratica la costruzione della successione termina dopo un numero finito di passi determinato in base alla verifica di opportuni *criteri di arresto*. La qualità e l'efficienza di un metodo iterativo è pertanto determinata dalle proprietà di convergenza della successione generata.

Una tecnica generale per derivare un metodo iterativo si basa sulla decomposizione additiva $A = M - N$ con $M$ matrice invertibile. Si ha allora

$$A\boldsymbol{x} = \boldsymbol{b} \iff (M - N)\boldsymbol{x} = \boldsymbol{b} \iff \boldsymbol{x} = M^{-1}N\boldsymbol{x} + M^{-1}\boldsymbol{b}.$$

Posto dunque $P = M^{-1}N$ detta *matrice di iterazione* e $q = M^{-1}\boldsymbol{b}$ si ottiene che

$$A\boldsymbol{x} = \boldsymbol{b} \quad \iff \quad \boldsymbol{x} = P\boldsymbol{x} + \boldsymbol{q},$$

ovvero $\boldsymbol{x}$ è soluzione del sistema lineare se e soltanto se

$$g(\boldsymbol{x}) = \boldsymbol{x}, \quad g\colon \mathbb{R}^n \to \mathbb{R}^n,\ g(\boldsymbol{z}) = P\boldsymbol{z} + \boldsymbol{q}. \tag{6.1}$$

Per la soluzione del problema *di punto fisso* (10.1) vale il seguente risultato.

**Teorema 6.1.1.** Dato $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$ sia $\boldsymbol{x}^{(k+1)} = g(\boldsymbol{x}^{(k)})$, $k \geq 0$. Se $\lim_{k\to+\infty} \boldsymbol{x}^{(k+1)} = \boldsymbol{x}$ allora $g(\boldsymbol{x}) = \boldsymbol{x}$ e dunque $A\boldsymbol{x} = \boldsymbol{b}$.

*Dimostrazione.* Segue dalla continuità di $g$. Vale infatti

$$0 \leq \| g(\boldsymbol{x}^{(k)}) - g(\boldsymbol{x}) \| = \| P\boldsymbol{x}^{(k)} - P\boldsymbol{x} \| \leq \| P \| \| \boldsymbol{x}^{(k)} - \boldsymbol{x} \|, \quad k \geq 0.$$

Quindi

$$\boldsymbol{x} = \lim_{k \to +\infty} \boldsymbol{x}^{(k+1)} = \lim_{k \to +\infty} g(\boldsymbol{x}^{(k)}) = g(\boldsymbol{x}).$$

□

Questo risultato motiva l'introduzione del seguente metodo iterativo per la risoluzione del sistema lineare $A\boldsymbol{x} = \boldsymbol{b}$ con $A \in \mathbb{R}^{n \times n}$ matrice invertibile:

$$\begin{cases} \boldsymbol{x}^{(0)} \in \mathbb{R}^n \\ \boldsymbol{x}^{(k+1)} = P\boldsymbol{x}^{(k)} + \boldsymbol{q}, \quad k \geq 0, \end{cases} \tag{6.2}$$

con

$$P = M^{-1}N, \quad \boldsymbol{q} = M^{-1}\boldsymbol{b}, \quad A = M - N.$$

Si osservi che (6.2) può essere scritto in maniera formalmente (ma non computazionalmente) equivalente come segue

$$\begin{cases} \boldsymbol{x}^{(0)} \in \mathbb{R}^n \\ M\boldsymbol{x}^{(k+1)} = N\boldsymbol{x}^{(k)} + \boldsymbol{b}, \quad k \geq 0. \end{cases} \tag{6.3}$$

**Definizione 6.1.1.** Il metodo (6.2) ((6.3)) si dice *convergente* se la successione generata dal metodo per ogni scelta del punto iniziale $\boldsymbol{x}^{(0)}$ converge alla soluzione $\boldsymbol{x} = A^{-1}\boldsymbol{b}$ del sistema lineare.

Il seguente risultato fornisce una condizione sufficiente per la convergenza del metodo (6.2).

**Teorema 6.1.2.** Il metodo (6.2) è convergente se esiste una norma matriciale indotta da una norma vettoriale $\| \cdot \|$ su $\mathbb{R}^n$ tale per cui $\| P \| < 1$.

*Dimostrazione.* Dalle relazioni

$$\boldsymbol{x}^{(k+1)} = P\boldsymbol{x}^{(k)} + \boldsymbol{q}, \quad \boldsymbol{x} = P\boldsymbol{x} + \boldsymbol{q},$$

segue che

$$\boldsymbol{e}^{(k+1)} = \boldsymbol{x}^{(k+1)} - \boldsymbol{x} = P(\boldsymbol{x}^{(k)} - \boldsymbol{x}) = P\boldsymbol{e}^{(k)}, \quad k \geq 0,$$

e quindi

$$\boldsymbol{e}^{(k+1)} = P^{k+1}\boldsymbol{e}^{(0)}, \quad k \geq 0.$$

Passando alla norma vettoriale (che induce la norma matriciale per cui $\| P \| < 1$) si ha

$$\| \boldsymbol{e}^{(k+1)} \| = \| P^{k+1}\boldsymbol{e}^{(0)} \| \leq \| P^{k+1} \| \| \boldsymbol{e}^{(0)} \|,$$

da cui

$$0 \leq \| \boldsymbol{e}^{(k+1)} \| \leq \| P \|^{k+1} \| \boldsymbol{e}^{(0)} \|,$$

da cui per il teorema del confronto segue che $\forall \boldsymbol{e}^{(0)}$ o, equivalentemente, $\forall \boldsymbol{x}^{(0)}$

$$\lim_{k \to +\infty} \| \boldsymbol{e}^{(k+1)} \| = \lim_{k \to +\infty} \| \boldsymbol{x}^{(k+1)} - \boldsymbol{x} \| = 0.$$

□

Il seguente risultato descrive una condizione necessaria per la convergenza del metodo (6.2). Ricordiamo che il *raggio spettrale* di una matrice $B \in \mathbb{R}^{n\times n}$ è definito come $\rho(B) = \max_i |\lambda_i|$, $\lambda_1, \ldots, \lambda_n$ autovalori di $B$.

**Teorema 6.1.3.** Se il metodo (6.2) è convergente allora $\rho(P) < 1$.

*Dimostrazione.* Sia $\lambda$ tale che $|\lambda| = \rho(P)$ e $\boldsymbol{v}$ un corrispondente autovettore di $P$, i.e., $P\boldsymbol{v} = \lambda\boldsymbol{v}$, $\boldsymbol{v} \neq \boldsymbol{0}$. Sia $\boldsymbol{x}^{(0)} = \boldsymbol{x} + \boldsymbol{v}$ con $\boldsymbol{x} = A^{-1}\boldsymbol{b}$ soluzione del sistema lineare $A\boldsymbol{x} = \boldsymbol{b}$. La successione generata dal metodo (6.2) con punto iniziale $\boldsymbol{x}^{(0)}$ è convergente ad $\boldsymbol{x}$. D'altra parte si ha

$$\boldsymbol{e}^{(k+1)} = P^{k+1}\boldsymbol{e}^{(0)} = P^{k+1}\boldsymbol{v} = \lambda^{k+1}\boldsymbol{v},$$

da cui

$$\| \boldsymbol{e}^{(k+1)} \| = \| \lambda^{k+1}\boldsymbol{v} \| = |\lambda|^{k+1} \| \boldsymbol{v} \|,$$

e quindi

$$\lim_{k\to+\infty} |\lambda|^k = 0$$

che implica

$$|\lambda| < 1.$$

□

Dal Teorema di Hirsch segue che per ogni norma matriciale indotta vale

$$\rho(A) \leq \| A \|, \quad \forall A \in \mathbb{R}^{n\times n}$$

per cui la condizione sufficiente implica la condizione necessaria. Inoltre vale il seguente risultato di cui omettiamo la dimostrazione.

**Teorema 6.1.4.** Sia $A \in \mathbb{R}^{n\times n}$ con $\rho(A) < 1$. Allora esiste una norma matriciale indotta tale per cui $\| A \| < 1$.

Combinando tra loro i teoremi 6.1.2, 6.1.3 e 6.1.4 si perviene infine al seguente risultato.

**Teorema 6.1.5.** Condizione necessaria e sufficiente per la convergenza del metodo iterativo (6.2) è che $\rho(P) < 1$.

# Lezione 6.2: I Metodi di Jacobi e Gauss-Seidel.

Sia $A = (a_{i,j}) \in \mathbb{R}^{n\times n}$ invertibile con *elementi diagonali non nulli*, i.e.,

$$a_{i,i} \neq 0, \quad 1 \leq i \leq n. \tag{6.4}$$

Poniamo $A = D - L - U$ con $D = (d_{i,j})$, $L = (l_{i,j})$ e $U = (u_{i,j})$ definite come segue:

$$d_{i,j} = \begin{cases} a_{i,j} \text{ se } i = j; \\ 0 \ altrimenti, \end{cases}$$

$$l_{i,j} = \begin{cases} -a_{i,j} \text{ se } i > j; \\ 0 \ altrimenti, \end{cases}$$

$$u_{i,j} = \begin{cases} -a_{i,j} \text{ se } i < j; \\ 0 \ altrimenti. \end{cases}$$

Il *metodo iterativo di Jacobi* per la risoluzione del sistema lineare $A\boldsymbol{x} = \boldsymbol{b}$ è definito dal partizionamento

$$M = D, \quad N = L + U.$$

Il *metodo iterativo di Gauss-Seidel* per la risoluzione del sistema lineare $A\boldsymbol{x} = \boldsymbol{b}$ è definito dal partizionamento

$$M = D - L, \quad N = U.$$

Poichè per entrambi i metodi $M$ risulta triangolare inferiore con elementi diagonali di $A$ si ha che la condizione (6.4) garantisce *l'applicabilità* dei metodi. Sotto tale assunzione i metodi sono implementati nella formulazione (6.3). Per il metodo di Jacobi si ottiene per $i = 1, 2, \ldots, n$,

$$a_{i,i}\boldsymbol{x}_i^{(k+1)} = \boldsymbol{b}_i - \sum_{j=1, j\neq i}^{n} a_{i,j}x_j^{(k)} \ \rightarrow \ \boldsymbol{x}_i^{(k+1)} = \frac{\boldsymbol{b}_i - \sum_{j=1, j\neq i}^{n} a_{i,j}x_j^{(k)}}{a_{i,i}}.$$

Per il metodo di Gauss-Seidel si ottiene

$$\sum_{j=1}^{i} a_{i,j}\boldsymbol{x}_j^{(k+1)} = \boldsymbol{b}_i - \sum_{j=i+1}^{n} a_{i,j}x_j^{(k)}, \quad i = 1, 2, \ldots, n,$$

da cui

$$\boldsymbol{x}_i^{(k+1)} = \frac{\boldsymbol{b}_i - \sum_{j=1}^{i-1} a_{i,j}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{i,j}x_j^{(k)}}{a_{i,i}}, \quad i = 1, 2, \ldots, n.$$

I seguenti programmi MatLab prendono in input la matrice $A$, il vettore $\boldsymbol{b}$ ed una approssimazione $\boldsymbol{x}\_old$ di $\boldsymbol{x}$ e restituiscono in output la nuova approssimazione $\boldsymbol{x}\_new$ di $\boldsymbol{x}$ generata dal metodo corrispondente. Per il metodo di Gauss-Seidel $\boldsymbol{x}\_new$ è sovrascritto direttamente in $\boldsymbol{x}\_old$.

```
function [x_new] = jacobi_mio(A,b,x_old)
n=length(b);
for k=1:n
    s=0;
    for j=1:k-1
        s=s+A(k,j)*x_old(j);
    end
    for j=k+1:n
        s=s+A(k,j)*x_old(j);
    end
    x_new(k)=(b(k)-s)/A(k,k);
end
end

function [x_old] = gauss_seidel_mio(A,b,x_old)
n=length(b);
```

```
for k=1:n
    s=0;
    for j=1:k-1
        s=s+A(k,j)*x_old(j);
    end
    for j=k+1:n
        s=s+A(k,j)*x_old(j);
    end
    x_old(k)=(b(k)-s)/A(k,k);
end
end
```

Detto nnz($A$) il numero di elementi non nulli della matrice $A$ si osserva che una iterazione del metodo di Jacobi e di Gauss-Seidel applicati per la risoluzione del sistema lineare $A\boldsymbol{x} = \boldsymbol{b}$ costa nnz($A$) operazioni moltiplicative. Pertanto i metodi sono particolarmente interessanti per la risoluzione numerica di sistemi lineari sparsi (nnz($A$) $<< n^2$).

La risoluzione numerica del sistema lineare $A\boldsymbol{x} = \boldsymbol{b}$ con un metodo iterativo richiede la determinazione di un *criterio di arresto* che consenta di terminare l'elaborazione. Criteri usualmente utilizzati sono del tipo

$$\| \boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)} \| \leq tol;$$

$$\frac{\| \boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)} \|}{\| \boldsymbol{x}^{(k)} \|} \leq tol;$$

$$\| A\boldsymbol{x}^{(k+1)} - \boldsymbol{b} \| \leq tol;$$

$$\frac{\| A\boldsymbol{x}^{(k+1)} - \boldsymbol{b} \|}{\| \boldsymbol{x}^{(k+1)} \|} \leq tol,$$

dove *tol* indica una tolleranza prefissata, eventualmente combinati con una condizione sul numero massimo di iterazioni *max_iter* eseguite in modo da garantire comunque (anche in caso di non convergenza) la terminazione del programma. Per i criteri basati sulla valutazione dell'errore assoluto e relativo in norma si osserva che se $\rho(P) < 1$ allora

$$\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)} = \boldsymbol{x}^{(k+1)} - \boldsymbol{x} + \boldsymbol{x} - \boldsymbol{x}^{(k)} = (P - I_n)(\boldsymbol{x}^{(k)} - \boldsymbol{x}),$$

da cui

$$\| \boldsymbol{x}^{(k)} - \boldsymbol{x} \| \leq \| (P - I_n)^{-1} \| \| \boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)} \| \leq tol \, \| (P - I_n)^{-1} \| .$$

Per i criteri basati sulla valutazione del residuo $\boldsymbol{r}^{(k+1)} = A\boldsymbol{x}^{(k+1)} - \boldsymbol{b}$ si ha

$$\boldsymbol{r}^{(k+1)} = A\boldsymbol{x}^{(k+1)} - \boldsymbol{b} = A\boldsymbol{x}^{(k+1)} - A\boldsymbol{x} = A(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}),$$

da cui

$$\| \boldsymbol{x}^{(k)} - \boldsymbol{x} \| \leq \| A^{-1} \| \| \boldsymbol{r}^{(k+1)} \| \leq tol \, \| A^{-1} \| .$$

Il seguente programma Matlab implementa il metodo di Jacobi arrestandosi quando $\| \boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)} \|_\infty \leq tol$ o $k > max\_iter$.

```
function [x_new] = jacobi_solve(A,b,x_old, tol, max_iter)
err=+inf;
it=0;
while(err>tol && it<=max_iter)
x_new=jacobi_mio(A,b,x_old);
err=norm(x_new'-x_old, 'inf');
x_old=x_new';
it=it+1;
end
it
end
```

# Lezione 6.3: Convergenza dei Metodi di Jacobi e Gauss-Seidel.

Nello studio della convergenza dei metodi iterativi di Jacobi e Gauss-Seidel siamo interessati a condizioni esprimibili in termini di proprietà della matrice $A$ dei coefficienti piuttosto che della matrice di iterazione $P$ che non è esplicitamente disponibile. Tra queste proprietà particolarmente rilevante è la seguente.

**Definizione 6.3.1.** Una matrice $A = (a_{i,j}) \in \mathbb{R}^{n\times n}$ si dice *predominante diagonale* (per righe) se

$$|a_{i,i}| > \sum_{j=1, j\neq i}^{n} |a_{i,j}|, \quad 1 \leq i \leq n.$$

Per sistemi predominanti diagonali vale

**Teorema 6.3.1.** Sia $A = (a_{i,j}) \in \mathbb{R}^{n\times n}$ predominante diagonale. Allora:

1. $A$ è invertibile;
2. i metodi di Jacobi e Gauss-Seidel per la risoluzione di un sistema lineare $A\boldsymbol{x} = \boldsymbol{b}$ sono applicabili;
3. i metodi di Jacobi e Gauss-Seidel per la risoluzione di un sistema lineare $A\boldsymbol{x} = \boldsymbol{b}$ sono convergenti.

*Dimostrazione.* Dimostriamo le tre proprietà.

1. L'invertibilità di $A$ segue dal teorema di Gershgorin. Infatti vale

$$|0 - a_{i,i}| = |a_{i,i}| > \sum_{j=1, j\neq i}^{n} |a_{i,j}|, \quad 1 \leq i \leq n,$$

e dunque

$$0 \notin U_{i=1}^{n} K_i.$$

2. Per l'applicabilità si ha

$$|a_{i,i}| > \sum_{j=1, j\neq i}^{n} |a_{i,j}| \geq 0 \Rightarrow |a_{i,i}| \neq 0, \quad 1 \leq i \leq n.$$

3. Dimostriamo quindi la convergenza. Dalla relazione

$$\det(P - \lambda I_n) = \det\big(M^{-1}N - \lambda I_n\big) = \det(N - \lambda M) = (-1)^n \det(\lambda M - N),$$

segue che $\lambda \in \mathbb{C}$ è autovalore di $P$ se e soltanto se $\det(\lambda M - N) = 0$. Si assuma ora $\lambda \in \mathbb{C}$, $|\lambda| \geq 1$. Si mostra che la matrice $\lambda M - N$ è predominante diagonale. Si ha infatti che

$$|a_{i,i}| > \sum_{j=1, j\neq i}^{n} |a_{i,j}| = \sum_{j=1}^{i-1} |a_{i,j}| + \sum_{j=i+1}^{n} |a_{i,j}|, \quad 1 \leq i \leq n,$$

implica

$$|\lambda||a_{i,i}| > |\lambda| \sum_{j=1}^{i-1} |a_{i,j}| + |\lambda| \sum_{j=i+1}^{n} |a_{i,j}| \geq |\lambda| \sum_{j=1}^{i-1} |a_{i,j}| + \sum_{j=i+1}^{n} |a_{i,j}|, \quad 1 \leq i \leq n,$$

e quindi

$$|\lambda a_{i,i}| > \sum_{j=1}^{i-1} |\lambda a_{i,j}| + \sum_{j=i+1}^{n} |a_{i,j}|, \quad 1 \leq i \leq n,$$

e

$$|\lambda a_{i,i}| > \sum_{j=1}^{i-1} |a_{i,j}| + \sum_{j=i+1}^{n} |a_{i,j}|, \quad 1 \leq i \leq n.$$

Queste relazioni esprimono la predominanza diagonale della matrice $\lambda M - N$ ottenuta rispettivamente nel metodo di Gauss-Seidel e di Jacobi per $\lambda \in \mathbb{C}$, $|\lambda| \geq 1$. Ma per il punto (1) sappiamo che una matrice predominante diagonale è invertibile e dunque $\lambda \in \mathbb{C}$, $|\lambda| \geq 1$ allora $\det(\lambda M - N) \neq 0$ per Jacobi e Gauss-Seidel. Segue che per gli autovalori delle matrici di iterazione di questi metodi deve valere $|\lambda| < 1$ e dunque $\rho(P) < 1$ e dunque la convergenza segue dal teorema 6.1.5.

□

# Lezione 6.4: Raffinamento Iterativo.

Abbiamo visto che il processo di eliminazione gaussiana in aritmetica a precisione finita determina un fattore triangolare $\tilde{U}$ ed un fattore "psychologically lower triangular matrix" $\tilde{L}$ per cui si ha (vedi relazione (5.4))

$$\tilde{L}\tilde{U} = \tilde{A} = A + E,$$

con $\| E \|$ piccola. Per raffinare la soluzione $\tilde{\boldsymbol{x}}$ del sistema lineare $A\boldsymbol{x} = \boldsymbol{b}$ ottenuta calcolando in macchina $\tilde{U}\tilde{\boldsymbol{x}} = \tilde{L}^{-1}\boldsymbol{b}$ si può procedere come segue. Dalla relazione $A\boldsymbol{x} = \boldsymbol{b}$ segue che

$$A\boldsymbol{x} = \tilde{A}\boldsymbol{x} - E\boldsymbol{x} = \boldsymbol{b},$$

e quindi

$$\boldsymbol{x} = \tilde{A}^{-1}E\boldsymbol{x} + \tilde{A}^{-1}\boldsymbol{b},$$

che motiva il metodo iterativo

$$\boldsymbol{x}^{(0)} = \tilde{\boldsymbol{x}}, \quad \boldsymbol{x}^{(k+1)} = \tilde{A}^{-1}E\boldsymbol{x}^{(k)} + \tilde{A}^{-1}\boldsymbol{b}, \ k \geq 0.$$

Si osserva che il metodo può essere riscritto come

$$\boldsymbol{x}^{(0)} = \tilde{\boldsymbol{x}}, \quad \tilde{A}\boldsymbol{x}^{(k+1)} = (\tilde{A} - A)\boldsymbol{x}^{(k)} + \boldsymbol{b}, \quad k \geq 0,$$

da cui si ottiene

$$\boldsymbol{x}^{(0)} = \tilde{\boldsymbol{x}}, \quad \tilde{A}(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}) = \boldsymbol{b} - A\boldsymbol{x}^{(k)}, \quad k \geq 0.$$

Il lettore dovrebbe dimostrare che se $\| \tilde{A}^{-1}E \| < 1$ allora il metodo iterativo converge. Si consideri quindi la procedura seguente che utilizza il programma *gauss_pp* descritto nella nota precedente. Si osservi l'uso di variabili della classe `sym` definite attraverso il Symbolic Toolbox di MatLab.

```
function [x] = itera_ref(n)
a=invhilb(n);
b=zeros(n,1);
b(1)=1;
x=gauss_pp(a,b)
pause
for k=1:n
    r=double(b-a*sym(x,'f'));
    y=gauss_pp(a,r);
    x=double(sym(y, 'f')+sym(x, 'f'))
    pause
end
end
```

Si descriva la procedura implementata e si commentino i risultati ottenuti sperimentalmente per $n = 8, 12, 16$.

# Lezione 6.5: Esercizi.

**Esercizio 29.** L'algoritmo "GeneRank" è stato introdotto come modifica intuitiva dell'algoritmo "PageRank" per l'analisi di similarità tra geni. Denotato con $\{g_1, \ldots, g_n\}$ l'insieme di $n$ geni definiti mediante un set di valori caratteristici ricavati dal "database GO" (http://geneontology.org), diciamo che due geni sono connessi se condividono almeno un valore. In questo modo definiamo la matrice di connettività $W = (w_{i,j}) \in \mathbb{R}^{n\times n}$ con elementi

$$w_{i,j} = \begin{cases} 1 \text{ se } g_i \text{ e } g_j \text{ sono connessi,} \\ 0, \text{ altrimenti.} \end{cases}$$

Sia inoltre $D \in \mathbb{R}^{n\times n}$ la matrice diagonale con elementi diagonali $d_i$, $1 \leq i \leq n$, tali che

$$d_i = \begin{cases} \sum_{j=1}^{n} w_{i,j} \text{ se la riga è non nulla,} \\ 1, \text{ altrimenti.} \end{cases}$$

Il problema del "GeneRank" consiste nella risoluzione del sistema lineare

$$(I - \alpha W D^{-1})\boldsymbol{x} = (1 - \alpha)\boldsymbol{b}, \quad 0 \leq \alpha \leq 1, \quad \boldsymbol{b} \in \mathbb{R}^n.$$

1. Si generi la matrice $W$ con i comandi $W = \texttt{rand}(1000) - 0.8$, $W = \texttt{ceil}(W)$. Utilizzando il comando `spy` si visualizzi la matrice. Utilizzando il comando `nnz` si detemini l'indice di sparsità di $W$.

2. Si dimostri che la matrice $I - \alpha W D^{-1}$ è invertibile.

3. Si dimostri che il metodo di Jacobi applicato ad $I - \alpha W D^{-1}$ risulta convergente.

4. Scrivere $W = \texttt{sparse}(W)$ e $[i, j, w] = \texttt{find}(W)$. Descrivere l'output e commentare i vantaggi di una implementazione dell'iterazione di Jacobi utilizzando questa rappresentazione della matrice.

**Esercizio 30.** Si consideri la matrice $A_n = (a_{i,j}^{(n)}) \in \mathbb{R}^{n \times n}$ definita da

$$a_{i,j}^{(n)} = \begin{cases} 1 \text{ se } i = j, \\ \alpha \text{ se } i \neq j. \end{cases}$$

con $\alpha$ parametro positivo. Per $n = 4$ si ha

$$A_4 = \begin{bmatrix} 1 & \alpha & \alpha & \alpha \\ \alpha & 1 & \alpha & \alpha \\ \alpha & \alpha & 1 & \alpha \\ \alpha & \alpha & \alpha & 1 \end{bmatrix}.$$

1. Determinare condizioni su $\alpha$ sufficienti a garantire la predominanza diagonale.

2. Scrivendo $A_n = (1 - \alpha) I_n + \alpha \boldsymbol{e}\boldsymbol{e}^T$, con $\boldsymbol{e} = [1, \ldots, 1]^T$, si verifichi che per $0 < \alpha < 1$ la matrice $A$ risulta definita positiva.

3. Investigare sperimentalmente la convergenza del metodo di Jacobi e di Gauss-Seidel per la risoluzione di $A_n \boldsymbol{x} = \boldsymbol{e}$ con $n = 128$ e $\alpha = 1/2$.

**Esercizio 31.** Sia $A \in \mathbb{R}^{3 \times 3}$ definita da

$$A = \begin{bmatrix} 1 & 2 & -2 \\ 1 & 1 & 1 \\ 2 & 2 & 1 \end{bmatrix},$$

e siano $J$ e $G$ rispettivamente le matrici di iterazione del metodo di Jacobi e Gauss-Seidel associate ad $A$.

1. Si calcoli $J^3$ e si dica se il metodo di Jacobi applicato ad $A$ risulta convergente.

2. Si dica se il metodo di Gauss-Seidel applicato ad $A$ risulta convergente.

3. Scrivere una funzione MatLab che dato in input $k \in \mathbb{N}$ e $\boldsymbol{b} \in \mathbb{R}^{3k}$ implementa il metodo di Jacobi per la risoluzione del sistema lineare $A_k \boldsymbol{x} = \boldsymbol{b}$ con vettore iniziale $\boldsymbol{x}_0$ nullo arrestandosi quando $\| \boldsymbol{x}_{\ell+1} - \boldsymbol{x}_\ell \|_\infty \leq 1.0e-12$, dove

$$A_k = \begin{bmatrix} I_k & 2I_k & -2I_k \\ I_k & I_k & I_k \\ 2I_k & 2I_k & I_k \end{bmatrix}$$

con $I_k$ matrice identica di ordine $k$.

4. Riportare il numero di iterazioni effettuate con $\boldsymbol{b} = \texttt{ones}(3k, 1)$, $k = 3, 4, 5$.

5. Giustificare i risultati ottenuti.

**Esercizio 32.** Sia $A \in \mathbb{R}^{n\times n} = (a_{i,j})$, $n > 1$, definita da

$$a_{i,j} = \begin{cases} n \text{ se } i = j; \\ -1 \text{ se } i \neq j. \end{cases}$$

1. Si dimostri che $A$ è definita positiva.

2. Sia $J$ la matrice di iterazione del metodo di Jacobi applicato per la risoluzione di un sistema lineare $A\boldsymbol{x} = \boldsymbol{b}$. Sia $\{\boldsymbol{x}_j\}$ la successione generata dal metodo e $\boldsymbol{x}^*$ la soluzione del sistema. Si calcoli $\| J \|_\infty$ e si determini un intero $k = k(n)$ tale da aversi

$$\| \boldsymbol{e}_k \|_\infty / \| \boldsymbol{e}_0 \|_\infty \leq 2^{-32},$$

   con $\boldsymbol{e}_j = \boldsymbol{x}_j - \boldsymbol{x}^*$, $j \geq 0$.

3. Scrivere una funzione MatLab che dati in input $\boldsymbol{b} \in \mathbb{R}^n$, $\boldsymbol{x}_0 \in \mathbb{R}^n$ e $j \in \mathbb{N}$ restituisce in output l'approssimazione $\boldsymbol{x}_j$ generata dal metodo di Jacobi implementato in modo da non richiedere la memorizzazione della matrice $A$ ed avere un costo per iterazione lineare in $n$.

4. Noto che $A\boldsymbol{e} = \boldsymbol{e}$ con $\boldsymbol{e} = \texttt{ones}(n, 1)$ si calcoli

$$r_j = \| \boldsymbol{e}_j \|_\infty / \| \boldsymbol{e}_0 \|_\infty$$

   per $n = 32$, $\boldsymbol{b} = \boldsymbol{e}$, $\boldsymbol{x}_0 = \boldsymbol{0}$ e $j = 8, 16, k(32)$.

**Esercizio 33.** Sia $A \in \mathbb{R}^{(n+1)\times(n+1)}$, $n \geq 2$, definita da

$$A = \left[\begin{array}{c|ccc} 1 & 0 & \dots & 0 \quad \alpha \\ \hline 1 & & & \\ \vdots & & I_n & \\ 1 & & & \end{array}\right], \quad \alpha \in \mathbb{R}.$$

1. Determinare i valori del parametro reale $\alpha$ per cui $A$ risulta predominante diagonale.

2. Determinare la matrice di iterazione $G \in \mathbb{R}^{(n+1)\times(n+1)}$ del metodo di Gauss-Seidel applicato ad $A$.

3. Determinare i valori del parametro reale $\alpha$ per cui il metodo di Gauss-Seidel applicato ad $A$ risulta convergente.

4. Scrivere una funzione MatLab che dati in input $\boldsymbol{b} \in \mathbb{R}^{n+1}$, $\alpha, tol \in \mathbb{R}$ senza memorizzare esplicitamente le matrici implementa il metodo di Gauss-Seidel con vettore iniziale nullo per la risoluzione del sistema lineare $A\boldsymbol{x} = \boldsymbol{b}$ restituendo in output una approssimazione $\boldsymbol{x}_k$ che soddisfi $\| A\boldsymbol{x}_k - \boldsymbol{b} \|_\infty \leq tol$.

5. Valutare il costo computazionale di una iterazione dell'algoritmo.

6. Per $n \in \{63, 127\}$, $\boldsymbol{b} = [1\colon n+1]^T$, $\alpha = 1/2$, $tol = 2^{-34}$ riportare il numero di iterazioni effettuate dall'algoritmo.

**Esercizio 34.** Sia $A \in \mathbb{R}^{n\times n}$, $n \geq 2$, definita da

$$A = (a_{i,j}) = \begin{cases} 1 \text{ se } j = i; \\ \gamma_i \in \mathbb{R} \text{ se } j = i+1; \\ \gamma_n \in \mathbb{R} \text{ se } i = n, j = 1; \\ 0 \text{ altrimenti.} \end{cases},$$

Sia inoltre $M$ la parte triangolare superiore di $A$ ed $N = M - A$. Per $n = 3$ si ha

$$A = \begin{bmatrix} 1 & \gamma_1 & 0 \\ 0 & 1 & \gamma_2 \\ \gamma_3 & 0 & 1 \end{bmatrix}, \quad M = \begin{bmatrix} 1 & \gamma_1 & 0 \\ 0 & 1 & \gamma_2 \\ 0 & 0 & 1 \end{bmatrix}, \quad N = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -\gamma_3 & 0 & 0 \end{bmatrix}.$$

Per la risoluzione del sistema lineare $A\boldsymbol{x} = \boldsymbol{b}$ si considera il metodo iterativo definito dal partizionamento $A = M - N$.

1. Determinare la matrice di iterazione del metodo.
2. Dire motivando la risposta se le seguenti affermazioni sono vere o false:
   (a) Se $A$ è predominante diagonale allora il metodo iterativo è convergente.
   (b) Il metodo iterativo è convergente se e solo se $A$ è predominante diagonale.
   (c) Il metodo iterativo è convergente se e solo se vale $\prod_{i=1}^{n} |\gamma_i| < 1$.
3. Scrivere una funzione MatLab che dati in input $\boldsymbol{b} = (b_i) \in \mathbb{R}^n$, $\gamma_1, \ldots, \gamma_n \in \mathbb{R}$, $\boldsymbol{x}_0 \in \mathbb{R}^n$ e $tol \in \mathbb{R}$, implementa il metodo iterativo per la risoluzione del sistema lineare $A\boldsymbol{x} = \boldsymbol{b}$ con vettore iniziale $\boldsymbol{x}_0$ arrestandosi quando $\| \boldsymbol{x}_{k-1} - \boldsymbol{x}_k \|_\infty \leq tol$ e restituendo in uscita $\boldsymbol{x}_k$ e $k$. L'implementazione non deve richiedere memorizzazioni e/o inversioni di matrici e deve avere costo computazionale lineare per iterazione.
4. Per $n \in \{64, 128, 512\}$, $tol = 10^{-12}$, $\boldsymbol{x}_0 = \boldsymbol{0}$, $\gamma_i = i/(i+1)$, $b_i = 1$, $1 \leq i \leq n$, riportare il numero di iterazioni eseguite dal metodo. Osservato che il numero di iterazioni decresce al crescere della dimensione si giustifichi tale comportamento.

**Esercizio 35.** Sia $B \in \mathbb{R}^{n\times n}$, $n \geq 1$, generata dal comando MatLab

```
B=(a^2 +1)*eye(n) +  a*gallery('tridiag', n,-1,0,-1);
```

con $a$ parametro reale, dove il comando

```
gallery('tridiag', n,d,b,c)
```

costruisce una matrice tridiagonale di ordine $n$ con elementi sottodiagonali, diagonali e sopradiagonali uguali rispettivamente a $d, b$ e $c$.

1. Si dimostri che per $|a| \neq 1$ la matrice $B$ è invertibile.

2. Si dimostri che per $|a| \neq 1$ il metodo di Jacobi applicato a $B$ è convergente.

3. Per $|a| \neq 1$ determinare un numero $k = k(a)$ di iterazioni del metodo di Jacobi applicato per la risoluzione del sistema lineare $B\boldsymbol{x} = \boldsymbol{b}$ sufficienti a garantire
$$\frac{\| \boldsymbol{e}_k \|_\infty}{\| \boldsymbol{e}_0 \|_\infty} \leq 2^{-32}.$$

4. Scrivere una funzione MatLab che dati in input $a \in \mathbb{R}$, $n \in \mathbb{N}$, $\boldsymbol{b} \in \mathbb{R}^n$ implementa il metodo di Jacobi per la risoluzione del sistema lineare $B\boldsymbol{x} = \boldsymbol{b}$ con punto iniziale $\boldsymbol{x}_0 = \boldsymbol{0}$ arrestandosi quando $\| \boldsymbol{x}_k - \boldsymbol{x}_{k-1} \|_\infty \leq 2^{-32}$ o $k \geq n$. L'implementazione non deve richiedere memorizzazioni e/o inversioni di matrici e deve avere costo computazionale lineare per iterazione.

5. Per $n = 128$, $a = 1/10, 1, 10$ e $b = \texttt{ones}(n, 1)$ riportare il numero di iterazioni eseguite dal metodo.

**Esercizio 36.** Sia $T_n(a) \in \mathbb{R}^{n \times n}$, $n \geq 2$, la matrice tridiagonale simmetrica con elementi sottodiagonali e diagonali rispettivamente uguali a $-1$ e $a \in \mathbb{R}$. Per $n = 4$ si ha
$$T_4(a) = \begin{bmatrix} a & -1 & 0 & 0 \\ -1 & a & -1 & 0 \\ 0 & -1 & a & -1 \\ 0 & 0 & -1 & a \end{bmatrix}.$$

1. Si dica motivando la risposta se le seguenti proposizioni sono vere o false:
   (a) Se $a > 2$ allora $T_n(a)$ è definita positiva e il metodo di Jacobi applicato a $T_n(a)$ converge.
   (b) Se $a > 0$ allora $T_n(a)$ è definita positiva e il metodo di Jacobi applicato a $T_n(a)$ converge.

2. Si mostri che se $T_n(a)$ è definita positiva allora il metodo di Jacobi è applicabile.

3. Si mostri che se $T_n(a)$ è definita positiva allora $a > 2\dfrac{n-1}{n}$.

4. Scrivere una funzione MatLab che dati in input $n \in \mathbb{N}$ $a \in \mathbb{R}$, $tol \in \mathbb{R}$, $itmax \in \mathbb{N}$ e $b \in \mathbb{R}^n$, implementa il metodo di Jacobi con vettore iniziale nullo per la risoluzione del sistema $T_n(a)\boldsymbol{x} = \boldsymbol{b}$. Il metodo si arresta quando $\| x_k - x_{k-1} \|_\infty < tol$ o $k > itmax$, riportando in output l'approssimazione $\boldsymbol{x}_k$ ed il numero di iterazioni $k$. L'implementazione non deve richiedere la memorizzazione esplicita della matrice.

5. Valutare il costo computazionale dell'algoritmo.

6. Per $tol = 1.0e{-}8$, $n = 128$, $itmax = n^2$, $\boldsymbol{b} = \texttt{ones}(n, 1)$, $a = 2{+}2/n, 2, 2{-}2/n$ riportare il valore di $k$ restituito dal programma.

**Esercizio 37.** Sia $A = (a_{i,j}) \in \mathbb{R}^{n \times n}$ definita da
$$a_{i,j} = \begin{cases} 2 \text{ se } i = j; \\ -1 \text{ se } i = j - 1; \\ -\alpha \text{ se } i = n, j = 1; \\ 0 \text{ altrimenti.} \end{cases}$$

1. Si determini i valori di $\alpha$ per cui $A$ risulta predominante diagonale. Per $\alpha = 2$ si calcoli la matrice $A_1$ ottenuta a partire da $A$ con un passo di eliminazione gaussiana e si dimostri che $A_1$ e quindi $A$ sono invertibili.

2. Si consideri il metodo iterativo $M = \text{triu}(A)$, $N = M - A$ per la risoluzione di un sistema lineare $A\boldsymbol{x} = \boldsymbol{b}$. Si determini i valori di $\alpha$ per cui il metodo è convergente.

3. Scrivere una funzione MatLab che dati in input $\boldsymbol{b} \in \mathbb{R}^n$, $\alpha \in \mathbb{R}$ e $\boldsymbol{x}_0 \in \mathbb{R}^n$ esegue un'iterazione del metodo restituendo in uscita $\boldsymbol{x}_1$. Se ne valuti il costo computazionale.

**Esercizio 38.** Sia $A \in \mathbb{R}^{n\times n}$ definita positiva, $\boldsymbol{b} \in \mathbb{R}^n$ e $\tau \neq 0$, $\tau \in \mathbb{R}$.

1. Si dimostri che $\boldsymbol{x} \in \mathbb{R}^n$ risolve

$$\boldsymbol{x} = (I - \tau A)\boldsymbol{x} + \tau \boldsymbol{b},$$

   se e solo se $A\boldsymbol{x} = \boldsymbol{b}$.

2. Si dimostri che il metodo iterativo

$$\boldsymbol{x}_{k+1} = (I - \tau A)\boldsymbol{x}_k + \tau \boldsymbol{b}, \quad k \geq 0,$$

   è convergente se $0 < \tau < 2/\lambda_{max}$, dove $\lambda_{max}$ indica l'autovalore più grande di $A$.

**Esercizio 39.** Sia $A \in \mathbb{R}^{2n\times 2n}$ definita da

$$A = \begin{bmatrix} L & I_2 & & \\ & \ddots & \ddots & \\ & & L & I_2 \\ I_2 & & & L \end{bmatrix},$$

con

$$L = \begin{bmatrix} 2 & 0 \\ -1 & 2 \end{bmatrix},$$

e $I_2$ matrice identità di ordine 2.

1. Si determini la matrice di iterazione $G$ del metodo di Gauss-Seidel applicato per la risoluzione del sistema lineare $A\boldsymbol{x} = \boldsymbol{b}$.

2. Si mostri che il metodo iterativo è convergente.

3. Si determini un valore di $\ell \in \mathbb{N}$ tale da aversi

$$\frac{\| \boldsymbol{x}_\ell - \boldsymbol{x} \|_1}{\| \boldsymbol{x}_0 - \boldsymbol{x} \|_1} \leq 2^{-32},$$

   con $\boldsymbol{x}$ soluzione del sistema lineare.

4. Si scriva una funzione MatLab che dati $n$, $\boldsymbol{b}$ ed $\boldsymbol{x0}$ calcola l'iterata $\boldsymbol{x1}$ generata dal metodo.

5. Si determini il costo computazionale di un'iterazione del metodo.

# Capitolo 7

# Calcolo di Autovalori ed Autovettori: Il Metodo delle Potenze

## Lezione 7.1: Generalità sul Metodo delle Potenze.

Il problema del calcolo di un autovalore e del corrispondente autovettore di una matrice $A \in \mathbb{C}^{n\times n}$ è affrontato mediante lo sviluppo di tecniche iterative che generano successioni $\{\lambda_k\}$ e $\{\boldsymbol{v}^{(k)}\}$ in modo da aversi

$$lim_{k\to+\infty}\lambda_k = \lambda, \quad lim_{k\to+\infty}\boldsymbol{v}^{(k)} = \boldsymbol{v},$$

con

$$A\boldsymbol{v} = \lambda\boldsymbol{v}, \quad \boldsymbol{v} \neq \boldsymbol{0}.$$

Tra queste tecniche si annovera *il metodo o i metodi delle potenze*. Nella forma più semplice essi generano a partire da un vettore iniziale $\boldsymbol{x}^{(0)}$, $\boldsymbol{x}^{(0)} \neq \boldsymbol{0}$, una sequenza di vettori $\{\boldsymbol{x}^{(k)}\}$ in accordo alla relazione

$$\begin{cases} \boldsymbol{z}^{(k)} = A\boldsymbol{x}^{(k)}; \\ \boldsymbol{x}^{(k+1)} = \frac{\boldsymbol{z}^{(k)}}{\beta_k}, \quad k \geq 0, \end{cases} \tag{7.1}$$

con $\beta_k \in \mathbb{C}$ detto *fattore di scala o di normalizzazione*. Strategie usuali per la scelta del parametro sono $\beta_k = 1$, $k \geq 0$, che corrisponde al *metodo non normalizzato* e $|\beta_k| = \| \boldsymbol{z}^{(k)} \|$, $k \geq 0$, che corrisponde ai *metodi con normalizzazione*. Si osserva che se $\boldsymbol{z}^{(j)} \neq \boldsymbol{0}$ per $0 \leq j \leq k-1$, allora vale

$$\boldsymbol{x}^{(k)} = \frac{\boldsymbol{z}^{(k-1)}}{\beta_{k-1}} = \frac{A\boldsymbol{x}^{(k-1)}}{\beta_{k-1}} = \frac{A\boldsymbol{z}^{(k-2)}}{\beta_{k-2}\beta_{k-1}} = \frac{A^2\boldsymbol{x}^{(k-2)}}{\beta_{k-2}\beta_{k-1}},$$

da cui

$$\boldsymbol{x}^{(k)} = \frac{A^k\boldsymbol{x}^{(0)}}{\prod_{j=0}^{k-1}\beta_j}, \quad k \geq 0, \tag{7.2}$$

che motiva il nome attribuito al metodo.

Un'analisi delle proprietà di convergenza delle sequenze di vettori generate dal metodo (7.1) può essere condotta sotto le seguenti assunzioni.

1. La matrice $A \in \mathbb{C}^{n\times n}$ è assunta *diagonalizzabile*.

2. Gli autovalori di $A$ possono essere ordinati in modo da aversi

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \ldots \geq |\lambda_n|.$$

   L'autovalore $\lambda_1$ è detto *autovalore dominante* di $A$.

3. Detti $\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(n)}$ gli autovettori corrispondenti agli autovalori $\lambda_1, \ldots, \lambda_n$ si ha che il vettore iniziale $\boldsymbol{x}^{(0)}$ rappresentato nella base di autovettori soddisfa

$$\boldsymbol{x}^{(0)} = \sum_{i=1}^{n} \alpha_i \boldsymbol{v}^{(i)} \text{ con } \alpha_1 \neq 0.$$

In particolare dalla relazione (7.2) per la proprietà (3) si ha

$$\boldsymbol{x}^{(k)} = \gamma_k^{-1} \sum_{i=}^{n} \alpha_i \lambda_i^k \boldsymbol{v}^{(i)}, \quad k \geq 0, \tag{7.3}$$

e

$$\boldsymbol{z}^{(k)} = \gamma_k^{-1} \sum_{i=}^{n} \alpha_i \lambda_i^{k+1} \boldsymbol{v}^{(i)}, \quad k \geq 0, \tag{7.4}$$

ove si è posto $\gamma_0 = 1$, $\gamma_k = \prod_{j=0}^{k-1} \beta_j$, $k \geq 1$. Ne discende il seguente.

**Teorema 7.1.1.** Se valgono (1), (2) e (3) allora le successioni dei vettori $\boldsymbol{x}^{(k)}$ e $\boldsymbol{z}^{(k)}$ generate dal metodo (7.1) sono ben definite ed inoltre si ha

$$\lim_{k\to+\infty} \frac{\gamma_k \boldsymbol{x}^{(k)}}{\lambda_1^k} = \alpha_1 \boldsymbol{v}^{(1)},$$

e

$$\lim_{k\to+\infty} \frac{\gamma_k \boldsymbol{z}^{(k)}}{\lambda_1^k} = \lambda_1 \alpha_1 \boldsymbol{v}^{(1)},$$

*Dimostrazione.* Da (7.3) e (7.4) segue che $\boldsymbol{x}^{(k)} \neq \boldsymbol{0}$ e $\boldsymbol{z}^{(k)} \neq \boldsymbol{0}$ per $k \geq 0$ essendo la componente di questi vettori lungo la direzione $\boldsymbol{v}^{(1)}$ non nulla. Per la relazione di limite si ha

$$\frac{\gamma_k \boldsymbol{x}^{(k)}}{\lambda_1^k} = \alpha_1 \boldsymbol{v}^{(1)} + \sum_{i=2}^{n} \alpha_i \left(\frac{\lambda_i}{\lambda_1}\right)^k \boldsymbol{v}^{(i)},$$

e quindi da $|\lambda_i/\lambda_1| < 1$ si ottiene

$$\lim_{k\to+\infty} \frac{\gamma_k \boldsymbol{x}^{(k)}}{\lambda_1^k} = \alpha_1 \boldsymbol{v}^{(1)} + \boldsymbol{0} = \alpha_1 \boldsymbol{v}^{(1)}.$$

La relazione per $\boldsymbol{z}^{(k)}$ si prova analogamente. □

Il teorema implica il seguente risultato.

**Teorema 7.1.2.** Nelle assunzioni (1), (2) e (3) se $\boldsymbol{v}_j^{(1)} \neq 0$ per un certo indice $j$ allora $\boldsymbol{x}_j^{(k)} \neq 0$ definitivamente, i.e., $\exists M \colon \boldsymbol{x}_j^{(k)} \neq 0 \ \forall k \geq M$.

*Dimostrazione.* Dal teorema precedente segue che

$$\lim_{k\to+\infty} \frac{\gamma_k \boldsymbol{x}_j^{(k)}}{\lambda_1^k} = \alpha_1 \boldsymbol{v}_j^{(1)}.$$

Posto dunque $0 < \epsilon < |\alpha_1 \boldsymbol{v}_j^{(1)}|$ si ha che $\exists M$ tale che

$$|\frac{\gamma_k \boldsymbol{x}_j^{(k)}}{\lambda_1^k} - \alpha_1 \boldsymbol{v}_j^{(1)}| \leq \epsilon, \quad \forall k \geq M,$$

e quindi

$$\boldsymbol{x}_j^{(k)} \neq 0 \quad \forall k \geq M.$$

□

Su questi risultati sono basate le tecniche di approssimazione dell'autovalore $\lambda_1$ e di un corrispondente autovettore.

# Lezione 7.2: Approssimazione dell'Autovalore Dominante.

Un primo risultato per l'approssimazione dell'autovalore $\lambda_1$ segue immediatamente.

**Teorema 7.2.1.** Siano soddisfatte le assunzioni (1), (2) e (3) e sia $\boldsymbol{v}_j^{(1)} \neq 0$ per un certo indice $j$. Allora vale

$$\lim_{k\to+\infty} \frac{\boldsymbol{z}_j^{(k)}}{\boldsymbol{x}_j^{(k)}} = \lambda_1.$$

*Dimostrazione.* Dal Teorema (7.1.2) segue che la sequenza di approssimazioni è ben definita da un certo punto in poi. Per la relazione di limite si ha

$$\lim_{k\to+\infty} \frac{z_j^{(k)}}{x_j^{(k)}} = \lim_{k\to+\infty} \frac{\gamma_k \lambda_1^k \boldsymbol{z}_j^{(k)}}{\gamma_k \lambda_1^k x_j^{(k)}} = \lambda_1.$$

□

La determinazione dell'indice $j$ può presentare alcune criticità computazionali. Nella seguente implementazione si sceglie $j$ come l'indice della prima componente di modulo massimo del vettore $\boldsymbol{x}^{(k)}$, $k \geq 0$.

```
function [lam, it] = power1(A, tol, maxiter)
n=length(A);
x=rand(n,1)+sqrt(-1)*rand(n,1);
lam=+inf; err=+inf;
it=0;
while(err>tol && it<=maxiter)
    [s,i]=max(abs(x));
    xi=x(i);
```

```
    x=A*x;
    lam1=x(i)/xi;
    err=abs(lam-lam1);
    lam=lam1;
    x=x/s;
    it=it+1;
end
```

Per ovviare alla determinazione dell'indice un approccio alternativo considera la seguente funzione detta *quoziente di Rayleigh*

$$r(\boldsymbol{x}) = \frac{\boldsymbol{x}^H A \boldsymbol{x}}{\boldsymbol{x}^H \boldsymbol{x}}, \quad \forall \boldsymbol{x} \neq \mathbf{0},\ \boldsymbol{x} \in \mathbb{C}^n.$$

Vale infatti il seguente.

**Teorema 7.2.2.** Sotto le assunzioni (1), (2) e (3) si ha

$$\lim_{k \to +\infty} r(\boldsymbol{x}^{(k)}) = \lambda_1.$$

*Dimostrazione.* Posto $\boldsymbol{w}^{(k)} = \dfrac{\gamma_k \boldsymbol{x}^{(k)}}{\lambda_1^k}$ si ha

$$r(\boldsymbol{x}^{(k)}) == \frac{\boldsymbol{x^{(k)}}^H A \boldsymbol{x}^{(k)}}{\boldsymbol{x^{(k)}}^H \boldsymbol{x}^{(k)}} = \frac{\boldsymbol{w^{(k)}}^H A \boldsymbol{w}^{(k)}}{\boldsymbol{w}^{(k)^H} \boldsymbol{w}^{(k)}} = r(\boldsymbol{w}^{(k)}),$$

e quindi per il teorema (7.1.1)

$$\lim_{k \to +\infty} r(\boldsymbol{x}^{(k)}) = \lim_{k \to +\infty} r(\boldsymbol{w}^{(k)}) = \lambda_1.$$

□

La seguente funzione implementa il metodo di approssimazione dell'autovalore dominante basato sul calcolo del quoziente di Rayleigh.

```
function [lam, it] = power2(A, tol, maxiter)
n=length(A);
x=rand(n,1)+sqrt(-1)*rand(n,1);
x=x/norm(x); z=A*x;
lam=+inf; err=+inf;
it=0;
while(err>tol && it<=maxiter)
    lam1=x'*z;
    err=abs(lam-lam1);
    lam=lam1;
    x=z/norm(z);
    z=A*x;
    it=it+1;
end
```

Il lettore confronti sperimentalmente i risultati restituiti dalle funzioni implementate per l'approssimazione dell'autovalore dominante delle matrici generate dal comando

```
A=gallery('tridiag', n, -1, 2, -1);
```

Il lettore dimostri che ler matrici così generate soddisfano le proprietà (1) e (2).

## Lezione 7.3: Approssimazione dell'Autovettore.

Per l'approssimazione di un autovettore relativo all'autovalore dominante valgono risultati analoghi.

**Teorema 7.3.1.** Siano soddisfatte le assunzioni (1), (2) e (3) e sia $\boldsymbol{v}_j^{(1)} \neq 0$ per un certo indice $j$. Allora vale

$$\lim_{k\to+\infty} \frac{\boldsymbol{x}^{(k)}}{\boldsymbol{x}_j^{(k)}} = \frac{\boldsymbol{v}^{(1)}}{\boldsymbol{v}_j^{(1)}}.$$

Per l'approssimazione mediante il quoziente di Rayleigh il risultato dipende dalla stategia di scelta del fattore di normalizzazione.

**Teorema 7.3.2.** Nelle assunzioni (1), (2) e (3) con $\beta_k = \| \boldsymbol{z}^{(k)} \|$ vale

$$\lim_{k\to+\infty} \| \boldsymbol{z}^{(k)} - r(\boldsymbol{x}^{(k)})\boldsymbol{x}^{(k)} \| = \lim_{k\to+\infty} \| A\boldsymbol{x}^{(k)} - r(\boldsymbol{x}^{(k)})\boldsymbol{x}^{(k)} \| = 0,$$

*Dimostrazione.* Poichè $\| x^{(k)} \| = 1$ si ha che

$$\lim_{k\to+\infty} \| A\boldsymbol{x}^{(k)} - r(\boldsymbol{x}^{(k)})\boldsymbol{x}^{(k)} \| = \lim_{k\to+\infty} \frac{\| A\boldsymbol{x}^{(k)} - r(\boldsymbol{x}^{(k)})\boldsymbol{x}^{(k)} \|}{\| x^{(k)} \|}.$$

D'altra parte posto come sopra $\boldsymbol{w}^{(k)} = \dfrac{\gamma_k \boldsymbol{x}^{(k)}}{\lambda_1^k}$ dal teorema 7.1.1 si ha

$$\lim_{k\to+\infty} \frac{\| A\boldsymbol{x}^{(k)} - r(\boldsymbol{x}^{(k)})\boldsymbol{x}^{(k)} \|}{\| x^{(k)} \|} = \lim_{k\to+\infty} \frac{\| A\boldsymbol{w}^{(k)} - r(\boldsymbol{w}^{(k)})\boldsymbol{w}^{(k)} \|}{\| w^{(k)} \|} = 0$$

per la continuità della norma. □

Dal teorema segue che nei metodi delle potenze con normalizzazione il vettore $\boldsymbol{x}^{(k)}$, $k \geq 0$, fornisce un'approssimazione dell'autovettore relativo all'autovalore dominante. La seguente modifica della funzione `power2` incorpora il calcolo dell'autovettore e restituisce in output una misura del residuo corrispondente.

```
function [lam, x, res,  it] = power2_mod(A, tol, maxiter)
n=length(A);
x=rand(n,1)+sqrt(-1)*rand(n,1);
x=x/norm(x); z=A*x;
lam=+inf; err=+inf;
it=0;
while(err>tol && it<=maxiter)
    lam1=x'*z;
    err=abs(lam-lam1);
    lam=lam1;
```

```
    x=z/norm(z);
    z=A*x;
    res=norm(z-lam1*x);
    it=it+1;
end
```

Il lettore confronti sperimentalmente sugli esempi introdotti precedentemente l'errore di approssimazione dell'autovalore e dell'autovedttore corrispondente.

# Lezione 7.4: Varianti del Metodo delle Potenze.

Il metodo delle potenze può essere modificato per permettere l'approssimazione dell'autovalore di modulo minino. Detti $\lambda_1, \ldots, \lambda_n$ gli autovalori di $A$ ed assunto che

$$|\lambda_1| \geq \ldots \geq |\lambda_{n-1}| > |\lambda_n| > 0,$$

allora per gli autovalori $\eta_i = 1/\lambda_i$ della matrice inversa $A^{-1}$ vale

$$|\eta_n| > |\eta_{n-1}| \geq \ldots \geq |\eta_1|.$$

Ne segue che il metodo detto *delle potenze inverse*

$$\begin{cases} A\boldsymbol{z}^{(k)} = \boldsymbol{x}^{(k)}; \\ \boldsymbol{x}^{(k+1)} = \frac{\boldsymbol{z}^{(k)}}{\beta_k}, \quad k \geq 0, \end{cases} \tag{7.5}$$

permette nelle assunzioni usuali di determinare approssimazioni di $\lambda_n$ e di un corrispondente autovettore. Analogamente se è disponibile una approssimazione iniziale $\eta$ di un autovalore $\lambda = \lambda_i$ di $A$ tale per cui

$$|\lambda - \eta| = |\lambda_i - \eta| < \min_{j \neq i} |\lambda_j - \eta|,$$

allora il il metodo detto *delle potenze inverse con shift*

$$\begin{cases} (A - \eta I_n)\boldsymbol{z}^{(k)} = \boldsymbol{x}^{(k)}; \\ \boldsymbol{x}^{(k+1)} = \frac{\boldsymbol{z}^{(k)}}{\beta_k}, \quad k \geq 0, \end{cases} \tag{7.6}$$

permette nelle assunzioni usuali di raffinare l'approssimazione di $\lambda_i$ ed eventualmente di un corrispondente autovettore.

Sotto opportune ipotesi il metodo delle potenze può essere esteso al calcolo di tutti gli autovalori dello spettro di $A$. Per fissare le idee assumiamo che $A$ sia una matrice simmetrica. È ben noto che in questo caso $A$ è diagonalizzabile ed inoltre autovettori relativi ad autovalori differenti sono ortogonali. Supponiamo inoltre che per gli autovalori $\lambda_1, \ldots, \lambda_n$ di $A$ valga

$$|\lambda_1| > |\lambda_2| > |\lambda_3| \geq \ldots \geq |\lambda_n|.$$

Sia $\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(n)}$ una base ortonormale di autovettori. Il lettore verifichi che la matrice $B = A - \lambda_1 \boldsymbol{v}^{(1)} {\boldsymbol{v}^{(1)}}^T$ ha autovalori $\lambda_2, \ldots, \lambda_n, 0$ e pertanto autovalore dominante $\lambda_2$. Ciò suggerisce la possibilità di procedere con l'approssimazione di $\lambda_2$ e di un corrispondente autovettore applicando il metodo delle potenze alla matrice

$$B = A - \tilde{\lambda}_1 \tilde{\boldsymbol{v}}^{(\mathbf{1})} {\tilde{\boldsymbol{v}}^{(\mathbf{1})}}^T,$$

dove $\tilde{\lambda}_1$ e $\tilde{\boldsymbol{v}}^{(\mathbf{1})}$ sono le approssimazioni calcolate per $\lambda_1$ e $\boldsymbol{v}^{(\mathbf{1})}$.

# Lezione 7.5: Esercizi.

**Esercizio 40.** Si consideri la matrice $A = (a_{i,j}) \in \mathbb{R}^{7\times 7}$ definita da

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

La matrice $A$ è la matrice di adiacenza di un grafo non orientato con 7 vertici indicati con $1, 2, \ldots, 7$ e tale che esiste un arco dal nodo $i$ al nodo $j$ se e solo se $a_{i,j} = a_{j,i} = 1$, $1 \leq i, j \leq 7$.

1. Rappresentare il grafo.
2. Utilizzando il teorema di Gerschgorin ottenere una maggiorazione per il modulo dell'autovalore dominante $\lambda_1$ di $A$. In particolare dimostrare che $\lambda_1 \leq n$, dove $n = 7$ indica il numero di vertici del grafo.
3. Determinare un'approssimazione dell'autovalore dominante utlizzando il metodo delle potenze applicato alla matrice $A$.
4. Supponiamo di voler assegnare un colore ad ogni vertice in modo che a vertici adiacenti (cioè connessi da un arco) siano assegnati colori differenti. Il minimo numero di colori sufficienti si chiama numero cromatico del grafo e si indica con $\chi$. È stato dimostrato che

   $$\frac{n}{n - \lambda_1} \leq \chi \leq \lambda_1 + 1.$$

   Utilizzando questo risultato dimostrare che $\chi \in \{2, 3\}$.
5. Determinare il valore di $\chi$

**Esercizio 41.** Si consideri la reazione chimica

$$x_1 KNO_3 + x_2 C \to x_3 K_2 CO_3 + x_4 CO + x_5 N_2.$$

Le equazioni di bilancio forniscono il seguente sistema lineare omogeneo per la determinazione dei valori incogniti $x_1, \ldots, x_5$:

$$\begin{cases} x_1 - 2x_3 = 0 \\ x_1 - 2x_5 = 0 \\ 3x_1 - 3x_3 - x_4 = 0 \\ x_2 - x_3 - x_4 = 0 \end{cases}$$

Sia $A \in \mathbb{R}^{4\times 5}$ la matrice dei coefficienti del sistema.

1. Si dimostri che $B = A^T A$ risulta semidefinita positiva.
2. Si dimostri che $B + \alpha I_5$ è definita positiva $\forall \alpha > 0$.

3. Si dimostri che se $A\boldsymbol{x} = \mathbf{0}$ e $\boldsymbol{x} \neq 0$ allora $(B + \alpha I_5)\boldsymbol{x} = \alpha\boldsymbol{x}$.

4. Si descriva un'approccio basato sul metodo delle potenze inverse per l'approssimazione del vettore $\boldsymbol{x}$.

5. Si implementi il metodo delle potenze inverse applicato alla matrice $B+I_5$ per l'approssimazione del vettore $\boldsymbol{x}$.

6. Si discutano approcci alternativi per la risoluzione del sistema lineare $A\boldsymbol{x} = 0$ basati sulle tecniche di fattorizzazione LU.

**Esercizio 42.** Si consideri il seguente programma MatLab

```
function[r]=sr(a, tol)
n=length(a);
%x=ones(n,1)/sqrt(n);
x=rand(n,1);
err=inf; r0=inf; kk=0;
while (err>tol)
    y=a*x;
    r=norm(y)/norm(x);
    x=y/norm(y);
    err=abs(r-r0);
    r0=r;
    kk=kk+1;
end
kk
```

con $a \in \mathbb{R}^{n\times n}$ e $tol \in \mathbb{R}^+$ dati di input.

1. Cosa si intende approssimare?

2. Si analizzi la convergenza del procedimento iterativo implementato.

3. Si analizzi sperimentalmente la convergenza per

   ```
   a=gallery('tridiag', 128, 1, -2, 1)
   ```

   e $tol = 1.0e-12$ utilizzando le due scelte del vettore iniziale descritte nel programma. Cosa si osserva? Spiegare il fenomeno.

4. Analizzare il costo computazionale dell'algoritmo.

5. Sostituire nel programma la norma euclidea con la norma infinito. Cosa si osserva per la convergenza?

**Esercizio 43.** La matrice di Lotka e Leslie modella la dinamica di una popolazione di individui divisi in fasce di età con diversi indici di natalità e mortalità. La matrice di ordine $n+1$ è definita da

$$A = \begin{bmatrix} m_1 & m_2 & \dots & \dots & m_{n+1} \\ s_1 & 0 & \dots & \dots & 0 \\ 0 & s_1 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & s_n & 0 \end{bmatrix},$$

con

- 1. $0 < s_i \leq 1$, $1 \leq i \leq n$;
  2. $0 \leq m_i \leq 1$, $1 \leq i \leq n+1$;
  3. $\gamma = \sum_{i=1}^{n+1} m_i > 0$;
  4. $\dfrac{s_i}{\gamma} \leq 1$, $1 \leq i \leq n$.

1. Applicando i teoremi di localizzazione alla matrice $B = (1/\gamma)A$ dimostrare che gli autovalori di $B$ hanno tutti modulo minore od uguale ad 1.

2. Scrivere una funzione MatLab che dati in input un vettore $\boldsymbol{s} = [s_1, \ldots, s_n]^T \in \mathbb{R}^n$, un vettore $\boldsymbol{m} = [m_1, \ldots, m_{n+1}]^T \in \mathbb{R}^{n+1}$, un naturale $j$, $1 \leq j \leq n+1$, ed un naturale $k > 0$, costruisce la matrice $A$ di parametri $m_i$ e $s_i$ e restituisce in output il vettore formato dalle $j$-esime componenti dei vettori generati in $k$ passi del metodo delle potenze applicato alla matrice $B$ a partire dal vettore $\boldsymbol{e} = [1, \ldots, 1]^T$.

3. Per una popolazione di pesci i parametri del modello di Lotka e Leslie sono:

| $i$ | $m_i$ | $s_i$ |
|---|---|---|
| 1 | 0 | 0.2 |
| 2 | 0.5 | 0.4 |
| 3 | 0.8 | 0.8 |
| 4 | 0.3 | – |

   Determinare l'approssimazione dell'autovalore dominante di $B$ ottenuta utilizzando la funzione implementata al punto precedente con valori $k = 20$ e $j = 1$.

**Esercizio 44.** Sia $p(z) = \prod_{i=1}^{n}(z - \lambda_i) = \sum_{i=0}^{n-1} a_i z^i + z^n$, $\lambda_i \neq \lambda_j$ se $i \neq j$, un polinomio monico di grado $n$ con zeri distinti e sia $C \in \mathbb{C}^{n \times n}$,

$$C = \begin{bmatrix} 0 & & & -a_0 \\ 1 & \ddots & & -a_1 \\ & \ddots & 0 & \vdots \\ & & 1 & -a_{n-1} \end{bmatrix},$$

la matrice "companion" associata a $p(z)$.

1. Si mostri che per $1 \leq j \leq n$ si ha

$$\left[1, \lambda_j, \ldots, \lambda_j^{n-1}\right] C = \lambda_j \left[1, \lambda_j, \ldots, \lambda_j^{n-1}\right].$$

2. Si dimostri che $C$ è diagonalizzabile.

3. Si dimostri che per gli zeri di $p(z)$ vale

$$\max_{1 \leq i \leq n} |\lambda_i| \leq \max\{|a_0|, 1 + |a_1|, \ldots, 1 + |a_{n-1}|\}.$$

4. Scrivere una funzione MatLab che dato in input che dati in input un vettore $\boldsymbol{a} = [a_0, \ldots, a_{n-1}]^T \in \mathbb{R}^n$ ed un naturale $k > 0$, restituisce in output il vettore formato dalle $n$-esime componenti dei vettori generati in $k$ passi del metodo delle potenze applicato alla matrice $C$ a partire dal vettore $\boldsymbol{e} = [1, 0 \ldots, 0]^T$.

5. Riportare le ultime 3 componenti del vettore generato dal programma con $\boldsymbol{a} = [-15, \ldots, -1]^T$ e $k = 48$.

6. Utilizzando queste componenti determinare le approssimazioni dello zero di modulo massimo del polinomio

$$p(z) = z^{15} - \sum_{i=0}^{14} (15 - i) z^i.$$

**Esercizio 45.** Il modello chiuso di Leontief assume che tutto il ricavo di un'industria sia utilizzato per acquistare i prodotti necessari per operare e che non via sia domanda esterna. Denotiamo con $S_1, \ldots S_n$ le industrie, con $p_j$, $1 \leq j \leq n$, il prezzo relativo del bene $x_j$ prodotto dall'industria $S_j$ e con $a_{i,j}p_i$ l'ammontare pagato dall'industria $S_j$ per il bene prodotto dall'industria $S_i$. Si ottiene il seguente sistema di equazioni lineari nelle incognite $p_1, \ldots, p_n$:

$$\begin{cases} p_1 = \sum_{i=1}^{n} a_{i,1} p_i \\ p_2 = \sum_{i=1}^{n} a_{i,2} p_i \\ \vdots \qquad \vdots \\ p_n = \sum_{i=1}^{n} a_{i,n} p_i \end{cases} \tag{7.7}$$

con la condizione $\sum_{j=1}^{n} a_{i,j} = 1$ per $1 \leq i \leq n$.

1. Si mostri che il sistema si riscrive equivalentemente come

$$[p_1, \ldots, p_n]\, A = [p_1, \ldots, p_n]$$

e

$$[p_1, \ldots, p_n]\, (A - I_n) = \mathbf{0}.$$

2. Si mostri che questo sistema è risolubile.

3. Si mostri che $\lambda = 1$ è un autovalore dominante di $A$.

4. Si confronti su alcuni esempi numerici l'accuratezza della soluzione determinata risolvendo il sistema lineare con un metodo diretto e calcolando $\boldsymbol{p}$ con il metodo delle potenze.

# Capitolo 8

# L'algoritmo di PageRanking

La ricerca di informazioni e l'analisi e classificazione dei dati presenti sul web conduce allo sviluppo di algoritmi innovativi per la risoluzione di problemi di algebra lineare di elevate dimensioni. Tra gli algoritmi di maggior successo ed impatto citiamo l'algoritmo di PageRank sviluppato dai fondatori di Google Larry Page e Sergey Brin quando erano studenti all'università di Stanford. L'algoritmo assegna un peso numerico ad ogni elemento (pagina) di un insieme di documenti uniti mediante collegamenti ipertestuali (hyperlink, o link), come ad esempio il World Wide Web, con lo scopo di quantificare la loro importanza relativa all'interno della raccolta.

Sia $G \in \mathbb{R}^{N\times N} = (g_{i,j})$ la matrice di connettività definita da

$$g_{i,j} = \begin{cases} 1 \text{ se esiste un link dalla pagina } j \text{ alla pagina } i; \\ 0 \text{ altrimenti,} \end{cases}$$

dove $N$ denota la cardinalità dell'insieme di documenti. Poniamo $c_j = \sum_{i=1}^{N} g_{i,j}$, $1 \leq j \leq N$, il numero di collegamenti che escono dalla pagina $j$. Assumiamo che la pagina $j$ distribuisca la sua importanza relativa al tempo $t = 0$ denotata con $p_j^{(0)}$ in modo uniforme alle pagine cui è collegata. Si ottiene allora che

$$p_i^{(1)} = \sum_{j=1}^{N} \frac{g_{i,j} p_j^{(0)}}{c_j}, \quad 1 \leq i, j \leq N. \tag{8.1}$$

Introdotta la matrice normalizzata $A \in \mathbb{R}^{N\times N} = (a_{i,j})$ definita da

$$a_{i,j} = \begin{cases} g_{i,j}/c_j \text{ se esiste un link dalla pagina } j \text{ alla pagina } i; \\ 0 \text{ altrimenti,} \end{cases}$$

possiamo riscrivere (8.1) nella forma

$$\boldsymbol{p}^{(1)} = A \cdot \boldsymbol{p}^{(0)},$$

da cui ponendo $\boldsymbol{p}^{(k)} = \left[p_1^{(k)}, \ldots, p_N^{(k)}\right]^T$

$$\boldsymbol{p}^{(k+1)} = A \cdot \boldsymbol{p}^{(k)}, \quad k \geq 0.$$

La sequenza di vettori $\{\boldsymbol{p}^{(k)}\}$ risulta pertanto generata dal metodo delle potenze senza normalizzazione applicato alla matrice $A$. Il vettore iniziale è generalmente scelto come $\boldsymbol{p}^{(0)} = (1/N)\texttt{ones}(N,1)$.

Per l'analisi della convergenza del metodo assumiamo che la matrice $G$ e quindi la matrice $A$ non abbiano colonne nulle. Si consideri il teorema di Gerschgorin applicato per colonne alla matrice $A$. Il generico cerchio $\mathcal{K}_\ell$ risulta definito da

$$\mathcal{K}_\ell = \{z \in \mathbb{C}\colon |z - a_{\ell,\ell}| \leq \sum_{i=1, i\neq\ell}^{N} |a_{i,\ell}|\}, \quad 1 \leq \ell \leq N.$$

Vale allora

$$a_{\ell,\ell} = 0, \quad \sum_{i=1, i\neq\ell}^{N} |a_{i,\ell}| = \sum_{i=1, i\neq\ell}^{N} \frac{g_{i,\ell}}{c_\ell} = 1,$$

da cui per il teorema di Gerschgorin segue che tutti gli autovalori $\lambda$ di $A$ soddisfano

$$|\lambda| \leq 1.$$

Inoltre posto $\boldsymbol{e} = \texttt{ones}(N,1)$ si ha

$$\boldsymbol{e}^T = \boldsymbol{e}^T \cdot A,$$

da cui segue che $\lambda = 1$ è autovalore di $A$. Sotto opportune ipotesi sulla struttura della matrice scalata delle connessioni $A$ segue che:

1. questo autovalore è dominante nel senso che gli autovalori di $A$ possono essere numerati in modo tale da aversi

$$1 = \lambda_1 = |\lambda_1| > |\lambda_2| \geq \ldots \geq |\lambda_N| \geq 0;$$

2. il Teorema 7.1.1 è applicabile e implica che

$$\lim_{k\to\infty} \boldsymbol{p}^{(k)} = \boldsymbol{v},$$

   con $A\boldsymbol{v} = \boldsymbol{v}$;

3. inoltre da

$$\boldsymbol{e}^T\boldsymbol{p}^{(k+1)} = \boldsymbol{e}^T A\boldsymbol{p}^{(k)} = \boldsymbol{e}^T\boldsymbol{p}^{(k)}$$

   si ha che la quantità $\boldsymbol{e}^T\boldsymbol{p}^{(k)} = \sum_{i=1}^{N} p_i^{(k)}$ si mantiene costante e quindi per il vettore limite vale

$$\sum_{i=1}^{N} v_i = 1, \quad v_i \geq 0,\ 1 \leq i \leq N.$$

Il vettore limite $\boldsymbol{v}$ è detto *vettore di pageranking* ed i suoi elementi $v_i$, $1 \leq i \leq N$, soddisfano $0 \leq v_i \leq 1$ e determinano l'importanza relativa della pagina $i$.

Al fine di estendere il calcolo in presenza di colonne nulle della matrice $A$ Page e Brin proposero l'applicazione del metodo delle potenze alla matrice

$$\widehat{A} = \alpha A + \frac{1-\alpha}{N}\boldsymbol{e}\boldsymbol{e}^T,$$

con $\alpha \in (0,1)$ ($\alpha = 0.85$ in Google). Il lettore può dimostrare che per la matrice $\widehat{A}$ continuano a valere le stesse proprietà di $A$. In particolare gli autovalori di $\widehat{A}$ appartengono al cerchio unitario, $\boldsymbol{e}^T\widehat{A} = \boldsymbol{e}^T$ e quindi $\lambda = 1$ è autovalore di $\widehat{A}$. Inoltre senza assunzioni aggiuntive si mostra che il metodo delle potenze senza normalizzazione con vettore iniziale $\boldsymbol{p}^{(0)} = (1/N)\texttt{ones}(N,1)$ applicato a $\widehat{A}$ genera una successione convergente ad un autovettore $\boldsymbol{v}$ di $\widehat{A}$ relativo all'autovalore 1 che soddisfa $\sum_{i=1}^{N} v_i = 1$. Infatti dalla relazione

$$\widehat{A}\boldsymbol{v} = \boldsymbol{v}$$

si ottiene

$$\alpha A\boldsymbol{v} + \frac{1-\alpha}{N}\boldsymbol{e}(\boldsymbol{e}^T\boldsymbol{v}) = \alpha A\boldsymbol{v} + \frac{1-\alpha}{N}\boldsymbol{e} = \boldsymbol{v}.$$

Questa relazione può essere riscritta come sistema lineare nella forma

$$(I - \alpha A)\boldsymbol{v} = \frac{1-\alpha}{N}\boldsymbol{e}.$$

Si verfica che la matrice dei coefficienti $B = I - \alpha A$ è predominante diagonale per colonne. Infatti si ha

$$\alpha \sum_{i=1, i\neq\ell}^{N} |a_{i,\ell}| \leq \alpha < 1, \quad 1 \leq \ell \leq N.$$

I metodi iterativi di Jacobi e Gauss-Seidel possono quindi essere applicati per la risoluzione del sistema lineare generando successioni convergenti al vettore $\boldsymbol{v}$ per ogni scelta del vettore iniziale $\boldsymbol{v}^{(0)}$. In particolare per il metodo di Jacobi si ottiene

$$M = I, \quad N = J = \alpha A.$$

La successione generata soddisfa

$$\boldsymbol{v}^{(k+1)} = \alpha A\boldsymbol{v}^{(k)} + \frac{1-\alpha}{N}\boldsymbol{e}, \quad k \geq 0.$$

Se $\boldsymbol{e}^T\boldsymbol{v}^{(0)} = \boldsymbol{e}^T\boldsymbol{p}^{(0)} = 1$ allora dalla relazione precedente segue che $\boldsymbol{e}^T\boldsymbol{v}^{(k)} = 1$ $\forall k \geq 0$ e quindi

$$\boldsymbol{v}^{(k+1)} = \alpha A\boldsymbol{v}^{(k)} + \frac{1-\alpha}{N}\boldsymbol{e}\boldsymbol{e}^T\boldsymbol{v}^{(k)} = \widehat{A}\boldsymbol{v}^{(k)}, \quad k \geq 0,$$

da cui segue che la successione generata dal metodo di Jacobi coincide con la successione generata dal metodo delle potenze applicato alla matrice $\widehat{A}$ con vettore iniziale $\boldsymbol{v}^{(0)}$. L'equivalenza tra i due metodi permette anche di stimare la convergenza. Dalla condizione sufficiente per la convergenza del metodo di Jacobi segue infatti che

$$\| \boldsymbol{v}^{(k)} - \boldsymbol{v} \|_1 \leq \alpha^k \| \boldsymbol{v}^{(0)} - \boldsymbol{v} \|_1, \quad k \geq 0.$$

# Capitolo 9

# Alcuni Problemi Numerici in Teoria dell'Approssimazione

## Lezione 9.1: Introduzione.

La teoria dell'approssimazione si occupa dell'approssimazione di funzioni e di dati ad esse associati (zeri, derivate, integrali, ...) mediante lo sviluppo di metodi ed algoritmi numerici. L'ambito di ricerca e di sviluppo di questi metodi è assai meno strutturato rispetto a quello dell'algebra lineare numerica. Gli algoritmi sono molteplici ed in generale il loro ambito di applicabilià è ristretto e soggetto alla verifica di opportune condizioni. Ciò è essenzialmente dovuto alla variabilità della struttura dei dati di ingresso. Nel seguito infatti faremo spesso riferimento a termini quali "data la funzione f" o "assegnata una funzione f". Sebbene formalmente accettabile questa terminologia nasconde diverse criticità computazionali. È infatti evidente che lo scenario varia radicalmente se la funzione $f$ è nota in forma esplicita o se la funzione $f$ è implicitamente disponibile mediante un algoritmo che restituisce il valore della funzione ed eventualmente delle sue derivate in ogni punto. Inoltre le proprietà di regolarità della funzione impattano drammaticamente sulla disponibilità e sulle prestazioni degli algoritmi di risoluzione.

Nel seguito ci occuperemo dei seguenti problemi:

1. *approssimazione degli zeri di una funzione*;
2. *approssimazione polinomiale di una funzione*;
3. *approssimazione dell'integrale definito di una funzione.*

In questa breve nota introduciamo questi problemi insieme ad alcune considerazioni computazionali riguardo il loro condizionamento.

# Lezione 9.2: Il Problema del Calcolo degli Zeri di una Funzione.

Il problema dell'approssimazione numerica degli zeri di una funzione reale di variabile reale viene generalmente formulato nel modo seguente.

**Problema 3.** Data una funzione $f\colon [a,b] \to \mathbb{R}$ si cercano (se esistono) valori della variabile per cui la funzione si annulla. Se $\xi \in [a,b]$ è un tale valore, i.e., $f(\xi) = 0$, allora $\xi$ è detto *zero della funzione* o, equivalentemente, *radice dell'equazione* $f(x) = 0$.

Un ben noto risultato che assicura l'esistenza di uno zero è il seguente *teorema di esistenza degli zeri.*

**Teorema 9.2.1.** Sia $f\colon [a,b] \to \mathbb{R}$, $f \in C^0([a,b])$, con $f(a)f(b) < 0$ allora $\exists\, \xi \in (a,b)$ tale che $f(\xi) = 0$.

Per l'approssimazione numerica degli zeri di una funzione si introducono tecniche iterative che a partire da un dato iniziale $x_0$ generano una successione $\{x_k\}$ di approssimazioni che sotto opportune ipotesi convergono ad uno zero della funzione, i.e.,

$$\lim_{k\to+\infty} x_k = \xi, \quad f(\xi) = 0.$$

IL caso non lineare presenta molteplici criticità rispetto al caso lineare. In particolare la convergenza dipende fortemente dalla scelta del punto iniziale e dalle proprietà della funzione ed eventualmente delle sue derivate. Qualunque metodo si intenda applicare, sarà quindi generalmente necessario effettuare uno studio preliminare della funzione in modo da *localizzare le eventuali radici*, i.e. determinare un intervallo che contenga una e una sola radice.

Le proprietà della funzione e delle sue derivate determinano anche il condizionamento del problema. Per semplicità di trattazione assumiamo che $f(x) = \phi(x) - d$ con $\phi\colon [a,b] \to \mathbb{R}$, $\phi \in C^2((a,b))$. Se $\xi \in (a,b)$ soddisfa $f(\xi) = 0$, $f'(\xi) \neq 0$ e $\hat{\xi} \in (a,b)$ soddisfa $\hat{f}(\hat{\xi}) = \phi(\hat{\xi}) - \hat{d} = 0$ con $\hat{d} = d + \epsilon$, $|\epsilon| \leq u$,allora vale

$$0 = \hat{f}(\hat{\xi}) = \phi(\hat{\xi}) - \hat{d} = \phi(\hat{\xi}) - \phi(\xi) - \epsilon.$$

Dallo sviluppo di Taylor

$$\phi(\hat{\xi}) \doteq \phi(\xi) + f'(\xi)(\hat{\xi} - \xi)$$

si ottiene quindi

$$|\hat{\xi} - \xi| \doteq \frac{|\epsilon|}{|f'(\xi)|},$$

da cui segue che

$$c = |f'(\xi)|^{-1}$$

è una misura del condizionamento del calcolo della radice semplice $\xi$ dell'equazione $f(x) = 0$. Quando $|f'(\xi)|$ è piccolo (idealmente zero) il problema diviene mal condizionato.

# Lezione 9.3: Il Problema dell'Approssimazione Polinomiale di una Funzione.

Il problema dell'approssimazione di una funzione si pone in diversi contesti applicativi. Per la valutazione di una funzione non razionale in macchina si richiede la determinazione di funzioni approssimanti di tipo polinomiale o razionale che possano essere valutate con un numero finito di operazioni aritmetiche. In molti contesti sperimentali una funzione incognita è nota mediante i valori (misure) assunti per differenti valori della variabile indipendente. In questi casi si cerca *un modello* ovvero una rappresentazione della funzione che ne consenta uno studio qualitativo e quantitativo. Infine lo sviluppo di appprossimazioni razionali e polinomiali di una funzione è alla base dei *metodi automatici per il calcolo di integrali e derivate*.

Di particolare interesse risulta il *problema dell'interpolazione polinomiale*.

**Problema 4.** Sia $\Pi_n$ lo spazio vettoriale dei polinomi a coefficienti reali di grado minore od uguale ad $n$. Assegnata $\phi_0(x), \ldots, \phi_n(x)$ una base di $\Pi_n$ e date $(x_i, y_i) \in \mathbb{R}^2$, $0 \leq i \leq n$, $n+1$ coppie di numeri reali con $x_i \neq x_j$ se $i \neq j$, si vuole determinare $\phi(x) \in \Pi_n$, i.e.,

$$\phi(x) = \sum_{i=0}^{n} \alpha_i \phi_i(x),$$

tale che $\phi(x_i) = y_i$ per $i = 0, \ldots, n$.

I punti $x_i$ sono detti *nodi dell'interpolazione*. Le condizioni $\phi(x_i) = y_i$ sono dette *condizioni di interpolazione*. Il polinomio $\phi(x)$ è detto *polinomio di interpolazione* sui punti $(x_i, y_i)$. Se $y_i = f(x_i)$, $0 \leq i \leq n$, e cioè $\phi(x)$ è determinato in modo da assumere sui nodi dell'interpolazione lo stesso valore di una funzione nota o incognita $f(x)$ allora $\phi(x)$ è detto *polinomio di interpolazione* sui nodi $x_i$ della funzione $f(x)$.

Da un punto di vista computazionale il calcolo del polinomio $\phi(x)$ si riduce alla risoluzione di un sistema lineare nelle variabili $\alpha_0, \ldots, \alpha_n$. Il condizionamento del calcolo del vettore dei coefficenti $\boldsymbol{\alpha}$ assegnata la base di $\Pi_n$ ed i punti dell'interpolazione viene pertanto misurato dal numero di condizione della matrice dei coefficenti del sistema lineare associato. Per quanto invece concerne il condizionamento della valutazione del polinomio $\phi(x)$ rispetto ad una perturbazione dei coefficenti si mostra facilmente che

$$|\sum_{i=0}^{n} \hat{\alpha}_i \phi_i(x) - \sum_{i=0}^{n} \alpha_i \phi_i(x)| \leq \| \hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha} \|_\infty \sum_{i=0}^{n} |\phi_i(x)|.$$

Se $a \leq x_1 < x_2 \ldots < x_n \leq b$ si ha che

$$\Delta = \max_{x \in [a,b]} \sum_{i=0}^{n} |\phi_i(x)|$$

definisce un indice o misura di condizionamento dell'approssimazione rispetto a perturbazione dei coefficienti. Il problema della stima del valore di $\Delta$ per una data configurazioni dei nodi è in generale di difficile soluzione. Per un'opportuna scelta della base di $\Pi_n$ è noto come *problema della determinazione della costante di Lebesgue*.

# Lezione 9.4: Il Problema del Calcolo dell'Integrale Definito di una Funzione.

Il problema del calcolo dell'integrale definito di una funzione è formulato come segue.

**Problema 5.** Data $f\colon [a,b] \to \mathbb{R}$, $f \in C^0([a,b])$, si vuole calcolare l'integrale definito

$$\mathcal{I}(f,a,b) = \int_a^b f(x)dx.$$

Il concetto di integrale è fondamentale nel calcolo e ha un'ampia applicazione in tutte le discipline scientifiche e tecnologiche. È associato al problema classico *del calcolo delle aree e dei volumi*. Metodi numerici per l'approssimazione di $\mathcal{I}(f,a,b)$ trovano inoltre applicazione nella risoluzione di *equazioni differenziali*. Il concetto di integrale è introdotto mediante un processo di limite che sarà approssimato mediante la costruzione di somme finite ottenute a partire da approssimazioni polinomiali della funzione integranda.

In generale il problema dell'integrazione numerica risulta "ben condizionato" come effetto delle proprietà di regolarizzazione del processo di integrazione. In particolare rispetto a perturbazioni della funzione integranda abbiamo che

$$|\mathcal{I}(f,a,b) - \mathcal{I}(\hat{f},a,b)| \leq (b-a) \max_{x\in[a,b]} |f(x) - \hat{f}(x)|.$$

# Capitolo 10

# Metodi Numerici per l'Approssimazione degli Zeri di una Funzione

## Lezione 10.1: Il Metodo di Bisezione.

Il metodo di bisezione è presumibilmente il più antico metodo noto per il calcolo degli zeri di una funzione. Sia $f\colon [a,b] \to \mathbb{R}$ con $f \in C^0([a,b])$ e $f(a)f(b) < 0$. Dal teorema di esistenza degli zeri segue che $\exists\ \xi \in [a,b]$ tale che $f(\xi) = 0$. Per la determinazione di un tale $\xi$ il *metodo di bisezione* genera sequenze di approssimazioni $\{a_k\}, \{b_k\}$ e $\{c_k\}$ definite come segue:

```
a(1)=a; b(1)=b;
for k>=1
    c(k)=(a(k)+b(k))/2;
    if (f(a(k))*f(c(k)<=0)
        a(k+1)=a(k);
        b(k+1)=c(k);
    else
        a(k+1)=c(k);
        b(k+1)=b(k);
    end
end
```

Il seguente teorema illustra le proprietà di convergenza delle successioni.

**Teorema 10.1.1.** Sia $f\colon [a,b] \to \mathbb{R}$ con $f \in C^0([a,b])$ e $f(a)f(b) < 0$. Per le successioni generate come sopra si ha

$$\lim_{k\to\infty} a_k = \lim_{k\to\infty} b_k = \lim_{k\to\infty} c_k = \xi \in [a,b]$$

con

$$f(\xi) = 0.$$

*Dimostrazione.* Si verifica facilmente che per costruzione $a_{k+1} \geq a_k$, $b_{k+1} \leq b_k$, $c_k \in [a_k, b_k] \subset [a,b]$, $0 \leq b_k - a_k \leq (b-a)/2^{k-1}$, $f(a_k)f(b_k) \leq 0$, $k \geq 1$. Ne

segue che esistono $\xi, \eta \in [a, b]$ tali che

$$\lim_{k\to\infty} a_k = \xi, \quad \lim_{k\to\infty} b_k = \eta.$$

Dal teorema del confronto segue che

$$\xi = \lim_{k\to\infty} a_k = \lim_{k\to\infty} b_k = \eta = \lim_{k\to\infty} c_k.$$

Per la continuità di $f$ si ha

$$\lim_{k\to\infty} f(a_k)f(b_k) = f(\xi)^2 \leq 0,$$

che implica

$$f(\xi) = 0.$$

□

Per l'implementazione in macchina del metodo di bisezione conviene fissare una tolleranza $\epsilon > 0$ e arrestare l'iterazione quando

$$b_{k+1} - a_{k+1} \leq \epsilon,$$

che garantisce

$$0 \leq \xi - a_{k+1} \leq \epsilon, \quad 0 \leq b_{k+1} - \xi \leq \epsilon, \quad 0 \leq |\xi - c_{k+1}| \leq \epsilon/2.$$

Poichè $0 \leq b_{k+1} - a_{k+1} \leq (b-a)/2^k$ si ha che la condizione risulta soddisfatta dopo

$$k \geq \lceil log_2(\frac{b-a}{\epsilon})\rceil,$$

iterazioni. Questo numero può essere significativamente elevato richiedendo molte valutazioni della funzione $f$. Il metodo di bisezione è quindi frequentemente utilizzato per fornire approssimazioni iniziali per procedure più efficienti. Il seguente programma implementa il metodo di bisezione in MatLab. La funzione è associata alla viariabile $f$ mediante la costruzione di un "anonymous function".

```
function c = bisection(f,a,b, tol)

if f(a)*f(b)>0
    disp('non inclusione');
    c='non trovato';
    return
else
    err = b-a; fa=f(a);
    while err > tol
        c = (a + b)/2;
        fc=f(c);
        if fa*fc<=0
            b = c;
        else
            a = c;
            fa=fc;
```

```
        end
        err=b-a;
    end
    c = (a + b)/2;
end
```

# Lezione 10.2: Metodi di Iterazione Funzionale.

Siano $f, g\colon [a,b] \to \mathbb{R}$. Le equazioni $f(x) = 0$ e $g(x) - x = 0$ si dicono *equivalenti* se $f(\xi) = 0 \iff g(\xi) - \xi = 0$ ovvero se $f(\xi) = 0 \iff g(\xi) = \xi$. In tal caso la radice $\xi$ dell'equazione $f(x) = 0$ è detta *punto fisso* della funzione $g(x)$. La riformulazione del problema della ricerca delle soluzioni di un'equazione come il problema della ricerca dei punti fissi di una funzione associata conduce all'introduzione dei metodi di iterazione funzionale del tipo

$$\begin{cases} x_0 \in [a,b]; \\ x_{k+1} = g(x_k), \quad k \geq 0. \end{cases} \tag{10.1}$$

Si ha infatti il seguente.

**Teorema 10.2.1.** Sia $g\colon [a,b] \to \mathbb{R}$, $g \in C^0([a,b])$. Se $x_k \in [a,b]$, $k \geq 0$, e $\lim_{k\to+\infty} x_k = \xi$ allora $\xi \in [a,b]$ e $g(\xi) = \xi$.

*Dimostrazione.* Dalla relazione di limite segue che $\xi \in [a,b]$ e per la continuità di $g$ che $g(\xi) = \xi$. □

Il teorema precedente chiarisce che la dinamica di (10.1) nel caso non lineare è più complicata in quanto dobbiamo assicurare che la successione generata a partire da un punto iniziale $x_0$ è ben definita e convergente.

**Definizione 10.2.1.** Sia $g\colon [a,b] \to \mathbb{R}$, $g(\xi) = \xi$, $\xi \in (a,b)$. Il metodo (10.1) si dice *localmente convergente* in $\xi$ se $\exists \rho > 0$ tale che $\forall x_0 \in [\xi-\rho, \xi+\rho] = I_\xi \subset [a,b]$ la successione generata dal metodo (10.1) soddisfa

1. $x_k \in I_\xi$ per ogni $k \geq 0$;
2. $\lim_{k\to+\infty} x_k = \xi$.

Un classico risultato che assicura la convergenza locale è il seguente *teorema del punto fisso.*

**Teorema 10.2.2.** Sia $g\colon [a,b] \to \mathbb{R}$, $g \in C^1([a,b])$, $g(\xi) = \xi$, $\xi \in (a,b)$. Se $\exists \rho > 0$ tale che $|g'(x)| < 1$ $\forall x \in [\xi - \rho, \xi + \rho] = I_\xi \subset [a,b]$ allora $\forall x_0 \in I_\xi$ la successione generata dal metodo (10.1) soddisfa

1. $x_k \in I_\xi$ per ogni $k \geq 0$;
2. $\lim_{k\to+\infty} x_k = \xi$.

*Dimostrazione.* Dal teorema di Weierstrass essendo $g'(x)$ continua e $I_\xi$ chiuso e limitato abbiamo $\lambda = \max_{x\in I_\xi} |g'(x)| < 1$. Si dimostra che la successione generata dal metodo (10.1) a partire da $x_0 \in I_\xi$ soddisfa

$$|x_k - \xi| \leq \lambda^k \rho, \quad k \geq 0, \tag{10.2}$$

da cui segue la proprietà (1)

$$|x_k - \xi| \leq \lambda^k \rho \leq \rho \quad \Rightarrow \quad x_k \in I_\xi,$$

e la proprietà (2) per il teorema del confronto

$$0 \leq |x_k - \xi| \leq \lambda^k \rho \quad \Rightarrow \quad \lim_{k \to +\infty} |x_k - \xi| = 0.$$

La dimostrazione di (10.2) procede per induzione su $k$. Per $k = 0$ da $x_0 \in I_\xi$ si ha

$$|x_0 - \xi| \leq \lambda^0 \rho = \rho.$$

Assumiamo quindi (10.2) vera fino all'indice $k$. Si ha allora per il teorema di Lagrange

$$|x_{k+1} - \xi| = |g(x_k) - g(\xi)| = |g'(\eta_k)(x_k - \xi)| = |g'(\eta_k)||(x_k - \xi)|, \quad |\eta_k - \xi| \leq |x_k - \xi|.$$

Per l'ipotesi induttiva segue che $\eta_k \in I_\xi$ e dunque

$$|x_{k+1} - \xi| = |g'(\eta_k)||(x_k - \xi)| \leq \lambda \lambda^k \rho = \lambda^{k+1} \rho.$$

□

Dal teorema segue il seguente.

**Teorema 10.2.3.** Sia $g\colon [a,b] \to \mathbb{R}$, $g \in C^1([a,b])$, $g(\xi) = \xi$, $\xi \in (a,b)$. Se $|g'(\xi)| < 1$ allora il metodo (10.1) è localmente convergente in $\xi$.

*Dimostrazione.* Sia $h\colon [a,b] \to \mathbb{R}$, $h(x) = |g'(x)| - 1$. Si ha che $h \in C^0([a,b])$, $h(\xi) < 0$ e dunque per il teorema della permanenza del segno $\exists I_\xi = [\xi - \rho, \xi + \rho] \subset [a,b]$ tale che $h(x) = |g'(x)| - 1 < 0 \ \forall x \in I_\xi$. La tesi allora segue dal teorema precedente. □

Di interesse computazionale risulta anche la caratterizzazione della convergenza. Nelle ipotesi del teorema 10.2.3 si assuma che $0 < |g'(\xi)| < 1$. Sia $\{x_k\}$ la successione generata dal metodo (10.1) a partire da $x_0 \in I_\xi$ intorno di convergenza. Se $x_k \neq \xi$, $k \geq 0$, allora

$$\frac{|x_{k+1} - \xi|}{|x_k - \xi|} = |g'(\eta_k)|, \quad |\eta_k - \xi| \leq |x_k - \xi|,$$

da cui segue

$$0 < \lim_{k \to +\infty} \frac{|x_{k+1} - \xi|}{|x_k - \xi|} = |g'(\xi)| < 1. \tag{10.3}$$

**Definizione 10.2.2.** Sia $\{x_k\}$ tale che $\lim_{k \to +\infty} x_k = \xi \in \mathbb{R}$, $x_k \neq \xi$, $k \geq 0$. Se vale (10.3) allora la successione è detta convergere *linearmente*.

Se $g'(\xi) = 0$ allora vale

$$\lim_{k \to +\infty} \frac{|x_{k+1} - \xi|}{|x_k - \xi|} = 0. \tag{10.4}$$

e la convergenza è più rapida.

**Definizione 10.2.3.** Sia $\{x_k\}$ tale che $\lim_{k\to+\infty} x_k = \xi \in \mathbb{R}$, $x_k \neq \xi$, $k \geq 0$. Se vale (10.4) allora la successione è detta convergere *superlinearmente*.

In particolare si distingue la seguente.

**Definizione 10.2.4.** Sia $\{x_k\}$ tale che $\lim_{k\to+\infty} x_k = \xi \in \mathbb{R}$, $x_k \neq \xi$, $k \geq 0$. Se vale

$$\lim_{k\to+\infty} \frac{|x_{k+1} - \xi|}{|x_k - \xi|^2} = \ell \in \mathbb{R}, \quad \ell \neq 0,$$

allora la successione è detta convergere *quadraticamente*.

Il teorema 10.2.3 fornisce una condizione sufficiente per la convergenza locale del metodo (10.1). La disamina del caso $|g'(\xi)| \geq 1$ risulta più involuta. Sia $g\colon [a,b] \to \mathbb{R}$, $g \in C^1([a,b])$, $g(\xi) = \xi$, $\xi \in (a,b)$ e $|g'(\xi)| > 1$. Sia $\{x_k\}$ una successione generata dal metodo (10.1) con $x_k \in [a,b]$, $x_k \neq \xi$, $k \geq 0$. Allora si conclude che la successione non converge a $\xi$.

> ♠ FAC Si assuma per assurdo che $\lim_{k\to+\infty} x_k = \xi$. Sia $I_\xi = [\xi-\rho, \xi+\rho] \subset [a,b]$ tale che $|g'(x)| \geq \lambda > 1$ $\forall x \in I_\xi$. Dalla definizione di limite si ha che $\exists \ell$ tale che $\forall k \geq \ell$ $x_k \in I_\xi$. Ma dal teorema di Lagrange segue che se $x_k \in I_\xi$ allora $\exists m > k$ tale che $x_m \notin I_\xi$ che contraddice la relazione di limite.

Ne segue che se l'equazione $g(x) = \xi$ ha $x = \xi$ come unica soluzione in $[a,b]$ allora il metodo (10.1) non è localmente convergente in $\xi$. A patto quindi di restringere eventualmente l'intervallo $[a,b]$ si conclude che se $g\colon [a,b] \to \mathbb{R}$, $g \in C^1([a,b])$, $g(\xi) = \xi$, $\xi \in (a,b)$ e $|g'(\xi)| > 1$ allora il metodo (10.1) non è localmente convergente in $\xi$. Se $|g'(\xi)| = 1$ si possono presentare situazioni di convergenza e di non convergenza richiedendo pertanto una valutazione specifica caso per caso.

L'implementazione del metodo (10.1) richiede la selezione di un opportuno *criterio di arresto* del tipo

$$|x_{k+1} - x_k| \leq tol, \quad \frac{|x_{k+1} - x_k|}{|x_{k+1}|} \leq tol.$$

Se $g \in C^1([a,b])$ allora

$$|x_{k+1} - x_k| = |x_{k+1} - \xi + \xi - x_k| = |g'(\xi_k) - 1||x_k - \xi|,$$

da cui si conclude che

$$|x_k - \xi| \leq \frac{tol}{|g'(\xi_k) - 1|}.$$

Ne segue che l'approssimazione restituita può essere scadente se $g'(\xi)$ è prossimo ad 1.

# Lezione 10.3: Il Metodo delle Tangenti.

Il più noto metodo di iterazione funzionale è il metodo delle tangenti o di Newton. Sia $f\colon [a,b] \to \mathbb{R}$, $f \in C^1([a,b])$, $f(\xi) = 0$, $\xi \in (a,b)$. Assegnata un'approssimazione iniziale $x_0 \in (a,b)$ di $\xi$ con $f'(x_0) \neq 0$ sia $y - f(x_0) = f'(x_0)(x - x_0)$

l'equazione della retta tangente al grafico della funzione nel punto $(x_0, f(x_0))$. Il punto di intersezione della retta con l'asse $y = 0$ ha ascissa

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

che si assume come nuova approssimazione di $\xi$. Iterando la procedura si ottiene il seguente metodo detto *metodo delle tangenti o di Newton*:

$$\begin{cases} x_0 \in [a, b]; \\ x_{k+1} = g(x_k) = x_k - \dfrac{f(x_k)}{f'(x_k)}, \quad k \geq 0. \end{cases} \tag{10.5}$$

Il primo risultato espone le proprietà di convergenza locale del metodo per l'approssimazione di *radici semplici*.

**Teorema 10.3.1.** Sia $f\colon [a, b] \to \mathbb{R}$, $f \in C^2([a, b])$, $f(\xi) = 0$, $f'(\xi) \neq 0$, $\xi \in (a, b)$. Allora il metodo (10.5) è localmente convergente in $\xi$, i.e., $\exists \rho > 0$ tale che $\forall x_0 \in [\xi - \rho, \xi + \rho] = I_\xi \subset [a, b]$ la successione generata dal metodo (10.5) soddisfa

1. $x_k \in I_\xi$ per ogni $k \geq 0$;
2. $\lim_{k \to +\infty} x_k = \xi$.

Se inoltre tale successione verifica $x_k \neq \xi$, $k \geq 0$, allora la convergenza è almeno quadratica, i.e.,

$$\lim_{k \to +\infty} \frac{|x_{k+1} - \xi|}{|x_k - \xi|^2} = \ell \in \mathbb{R}.$$

*Dimostrazione.* Da $f'(\xi) \neq 0$ per il teorema della permanenza del segno segue che $\exists I'_\xi = [\xi - \rho', \xi + \rho'] \subset [a, b]$ tale che $f'(x) \neq 0$ $\forall x \in I'_\xi$. Si verifica quindi che la funzione di iterazione $g\colon I'_\xi \to \mathbb{R}$, $g(x) = x - \dfrac{f(x)}{f'(x)}$ soddisfa $g \in C^1(I'_\xi)$ e $g'(x) = \dfrac{f(x)f''(x)}{(f'(x))^2}$. Poichè quindi $g'(\xi) = 0$ la prima parte del teorema segue dal teorema 10.2.3. Per la stima della velocità di convergenza dallo sviluppo di Taylor arrestato al secondo ordine si ottiene

$$0 = f(\xi) = f(x_k) + f'(x_k)(\xi - x_k) + \frac{f''(\eta_k)(\xi - x_k)^2}{2}, \quad |\eta_k - \xi| \leq |x_k - \xi|,$$

da cui

$$x_{k+1} - \xi = \frac{f''(\eta_k)(\xi - x_k)^2}{2f'(x_k)}, \quad |\eta_k - \xi| \leq |x_k - \xi|,$$

da cui si ricava per continuità di $f'(x)$ e $f''(x)$

$$\lim_{k \to +\infty} \frac{|x_{k+1} - \xi|}{|x_k - \xi|^2} = \left| \frac{f''(\xi)}{2f'(\xi)} \right| \in \mathbb{R}.$$

□

Di rilevante interesse computazionale per il metodo delle tangenti sono anche alcuni risultati detti di *convergenza in grande*. Nella seguente versione si assicura la convergenza per punti iniziali opportunamente scelti in un intervallo destro della soluzione. Un analogo risultato può essere formulato per un intervallo sinistro.

**Teorema 10.3.2.** Sia $f\colon [a,b] \to \mathbb{R}$, $f \in C^2([a,b])$, $f(\xi) = 0$, $\xi \in (a,b)$. Se $\exists \delta > 0$ tale che $\forall x \in (\xi, \xi+\delta] = I_\delta \subset [a,b]$ si ha

1. $f'(x) \neq 0$;
2. $f(x)f''(x) > 0$;

allora il metodo (10.5) con $x_0 \in I_\delta$ genera successioni convergenti ad $\xi$.

*Dimostrazione.* Si osserva che $f'(x)$ ha segno costante in $I_\delta$. Si assuma per fissare le idee che $f'(x) > 0$. Allora segue che $f(x) > 0$ e $f''(x) > 0$. Si ha allora che

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \leq x_0.$$

Inoltre

$$x_1 - \xi = g(x_0) - g(\xi) = g'(\eta_0)(x_0 - \xi) = \frac{f(\eta_0)f''(\eta_0)}{(f'(\eta_0))^2}(x_0 - \xi) > 0,$$

e quindi

$$\xi < x_1 \leq x_0 \leq \xi + \delta.$$

In modo analogo si dimostra per induzione che per i termini della successione vale

$$\xi < x_{k+1} \leq x_k \leq \xi + \delta, \quad k \geq 0.$$

Ne segue che

$$\lim_{k \to +\infty} x_k = \alpha \in [\xi, \xi + \delta],$$

e dunque

$$\alpha = g(\alpha) \quad \Rightarrow \quad f(\alpha) = 0 \quad \Rightarrow \quad \alpha = \xi.$$

□

# Lezione 10.4: Il Caso delle Equazioni Algebriche.

Un problema di rilevante interesse applicativo è il calcolo delle radici di un'equazione algebrica a coefficienti reali

$$f(x) = p(x) = p_n x^n + p_{n-1} x^{n-1} + \ldots + p_1 x + p_0 = 0, \quad p_i \in \mathbb{R}, \quad p_n \neq 0$$

È ben noto che l'equazione ammette $n$ radici eventualmente complesse contate con le loro molteplicità. Per la determinazione di alcune radici reali il metodo di Newton può essere utilizzato e richiede ad ogni iterazione una valutazione del polinomio e della sua derivata. Il seguente *algoritmo di Horner* è impiegato per il calcolo.

```
function [px,dx] = horner(p, x0)
 n1=length(p);
 n=n1-1;
 px = p(n+1);
 dx= 0;
   for k= n:-1:1
       dx= px + x0 * dx;
       px = p(k)+ x0 * px;
   end

end
```

Per l'approssimazione di tutte le radici reali e complesse dell'equazione è conveniente la riformulazione del problema in termini del calcolo degli autovalori di una opportuna matrice detta *matrice companion* associata con il polinomio $p(x)$. Si ha che posto

$$F(\boldsymbol{p}) = \begin{bmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & 0 & 1 \\ -\dfrac{p_0}{p_n} & \cdots & \cdots & -\dfrac{p_{n-1}}{p_n} \end{bmatrix},$$

allora

$$\det(xI_n - F(\boldsymbol{p})) = p(x)/p_n,$$

e dunque il problema del calcolo delle radici dell'equazione algebrica è ricondotto al calcolo degli autovalori della matrice $F(\boldsymbol{p})$. Questo approccio è seguito in MatLab ed il comando `roots` implementa la procedura descritta.

# Lezione 10.5: Esercizi.

**Esercizio 46.** Si consideri la seguente equazione nota come equazione di Keplero

$$f(x) = x - \epsilon \sin x - \eta = 0, \quad 0 < |\epsilon| < 1,\ \eta \in \mathbb{R}.$$

1. Posto $a = \eta - |\epsilon|$ e $b = \eta + |\epsilon|$ si mostri che
$$f(a) \leq 0, \quad f(b) \geq 0.$$
2. Si dimostri quindi che l'equazione di Keplero ha una sola soluzione reale $\xi$ e $\xi \in [a, b]$.
3. Si mostri che il metodo iterativo
$$\begin{cases} x_0 \in \mathbb{R} \\ x_k = \epsilon \sin x_{k-1} + \eta,\ k \geq 1, \end{cases}$$
è convergente a $\xi$ per ogni scelta del punto iniziale.
4. Si implementino in MatLab una funzione per la risoluzione dell'equazione mediante il metodo di bisezione ed una funzione per la risoluzione dell'equazione mediante il metodo di iterazione funzionale descritto al punto precedente. Si confronti sperimentalmente la convergenza dei due metodi.

**Esercizio 47.** La legge oraria di un corpo in caduta verticale soggetto alla resistenza dell'aria può essere modellata come

$$s(t) = s_0 - \alpha g t + \alpha^2 g(1 - e^{-t/\alpha}), \quad t, s_0, g, \alpha > 0.$$

1. Determinare il numero di soluzioni reali dell'equazione $s(t) = 0$.
2. Per ogni soluzione determinare un'intervallo di inclusione/separazione $[a, b]$.
3. Scrivere una funzione MatLab che dati in input $\alpha, g, s_0, a, b$ implementa il metodo di bisezione per l'approssimazione della soluzione dell'equazione $s(t) = 0$ in $[a, b]$.
4. Studiare la convergenza del metodo iterativo

$$t_{k+1} = \frac{s_0}{\alpha g} + \alpha(1 - e^{-t_k/\alpha}), \quad t_0 > 0,$$

per la risoluzione dell'equazione.

**Esercizio 48.** Si consideri l'equazione

$$x = g(x), \quad g(x) = -\frac{1}{2}x^3 - \frac{3}{2}x^2 + 1.$$

1. Si determini il numero di soluzioni reali dell'equazione.
2. Per l'approssimazione delle soluzioni reali dell'equazione si introducono i seguenti metodi iterativi:

$$x_{k+1} = g(x_k),$$

e

$$x_{k+1} = f(x_k), \quad f(x) = \frac{2 + 3x^2 + 2x^3}{2 + 6x + 3x^2}.$$

3. Studiare la convergenza locale dei metodi iterativi.
4. Dire se la successione generata dal primo metodo con $x_0 = 0$ risulta convergente.
5. Utilizzare il metodo delle potenze inverse per approssimare la soluzione $\alpha$ di modulo minimo dell'equazione.
6. Si mostri che il secondo metodo iterativo genera successioni convergenti ad $\alpha$ per ogni $x_0 \geq \alpha$.

**Esercizio 49.** Si consideri l'equazione

$$x = \phi_\omega(x), \quad \phi_\omega(x) = \frac{e^{-x} + \omega x}{1 + \omega}, \quad \omega \in \mathbb{R},\ \omega \neq -1. \tag{10.6}$$

1. Dimostrare che l'equazione ammette una sola soluzione reale $\alpha$ non dipendente da $\omega$ con $\alpha \in [1/2, 1]$.
2. Dimostrare che il metodo iterativo $x_{n+1} = \phi_0(x_n)$, $n \geq 0$, a partire da $x_0 = 1$ genera una successione convergente ad $\alpha$.

3. Dimostrare che il metodo iterativo $x_{n+1} = \phi_1(x_n)$, $n \geq 0$, a partire da $x_0 = 1$ genera una successione convergente ad $\alpha$.

4. Dimostrare che
$$|\phi'_1(\alpha)| < |\phi'_0(\alpha)|.$$

5. Scrivere una funzione MatLab che dati in input $\omega \in \mathbb{R}$, $\omega \neq -1$, $x_o \in \mathbb{R}$, $tol \in \mathbb{R}$ e $itmax \in \mathbb{N}$ calcola un'approssimazione di $\alpha$ generando la sequenza $x_{n+1} = \phi_\omega(x_n)$, $n \geq 0$. Il calcolo si arresta quando $|x_{n+1} - x_n| \leq tol$ o $n + 1 > itmax$ e la funzione restituisce in output il valore di $n + 1$ e la corrispondente approssimazione $x_{n+1}$.

6. Riportare il valore di $n+1$ e $x_{n+1}$ ottenuto assegnando ad $\omega$ i valori $0, \frac{1}{2}, 1$ con $x_0 = 1$, $tol = 1.0e - 14$ e $itmax = 100$.

7. Sia $\tilde{\alpha} = x_{n+1}$ l'approssimazione di $\alpha$ calcolata al passo precedente per $\omega = 1/2$. Riportare il valore di $n + 1$ e $x_{n+1}$ restituito dalla funzione con $\omega = \tilde{\alpha}$, $x_0 = 1$, $tol = 1.0e - 14$ e $itmax = 100$.

8. Commentare i risultati sperimentali ottenuti per i differenti valori di $\omega$.

**Esercizio 50.** Si consideri l'equazione
$$f(x) = x^2 - 4 \ \sin x = 0.$$

1. Si dimostri che esiste una sola soluzione $\alpha$ nell'intervallo $(0, \pi]$.

2. Si determini il numero di soluzioni reali dell'equazione.

3. Si determini un intervallo iniziale $[a, b]$ per l'approssimazione di $\alpha$ con il metodo di bisezione.

4. Scrivere una funzione MatLab che dati in input $a, b, tol$ implementa il metodo di bisezione applicato alla funzione $f(x)$ per l'approssimazione di $\alpha$ restituendo in output il punto medio del segmento di estremi $a_k$, $b_k$ tale per cui $b_k - a_k < tol$ e $b_{k-1} - a_{k-1} \geq tol$.

5. Riportare le approssimazioni fornite dal programma per $tol = 2^{-p}$, $p = 8, 16, 32$.

**Esercizio 51.** Per $n \in \mathbb{N}$, $n \geq 1$ sia
$$f_n(x) = \sum_{k=0}^{n} \frac{1}{k - x}.$$

1. Scrivere una funzione MatLab che dati in input $n$ ed $x \in \mathbb{R}$ ritorna in output il valore della funzione $f_n(x)$ e della sua derivata prima $f'_n(x)$ valutate in $x$.

2. Determinare il costo computazionale dell'algoritmo.

3. Mostrare che l'equazione $f_n(x) = 0$ ha una sola soluzione reale $\alpha_n$ nell'intervallo $(n - 1, n)$.

4. Per l'approssimazione di tale soluzione si considera il metodo iterativo

$$x_{k+1} = x_k - \frac{f_n(x_k)}{f_n'(x_k) - (f_n(x_k))^2}, \quad k \geq 0.$$

Scrivere una funzione MatLab che dato in input $x_0$, $tol$, $maxiter$ e $n$ utilizzando la funzione del punto 1) calcola i termini della successione generata dal metodo iterativo arrestandosi se $k > maxiter$ o $|x_{k+1} - x_k| \leq tol$.

5. Riportare i termini della successione genarati per $n = 11$, $x_0 = 11.1$, $maxiter = 100$ e $tol = 1.0e-04$.

6. Studiare la convergenza locale del metodo per l'approssimazione di $\alpha_n$.

**Esercizio 52.** Si consideri la funzione $g\colon \mathbb{R}^+ \to \mathbb{R}$ definita da

$$g(u) = -\frac{1}{2} + \int_0^u e^{-x^2}\,\mathrm{d}x.$$

È noto che $\lim_{u\to+\infty} g(u) = \dfrac{\sqrt{\pi}-1}{2}$ e che $g'(u) = e^{-u^2}$ per $u \geq 0$. Per la valutazione della funzione si può utilizzare il comando Matlab®:

```
q=integral(@(x)exp(-x.^2),0,u)-1/2
```

che restituisce il valore $q = g(u)$ per $u \geq 0$.

1. Utilizzando i comandi `integral` e `plot` di Matlab® tracciare un grafico della funzione $g$ per $0 \leq u \leq 4$.

2. Mostrare che l'equazione $g(u) = 0$ ha una sola soluzione reale positiva denotata con $\alpha$.

3. Mostrare che il metodo delle tangenti applicato per la risoluzione dell'equazione $g(u) = 0$ con punto iniziale $u_0 = 1/2$ genera una successione convergente alla soluzione.

4. Dire motivando la risposta se la convergenza della successione generata al punto precedente è almeno quadratica.

5. Scrivere una funzione MatLab che dati in input $tol$ e $u_0$ utilizzando il comando `integral` per valutare la funzione implementa il metodo delle tangenti per la risoluzione dell'equazione restituendo un valore $u_k$ tale da aversi $|u_k - u_{k-1}| \leq tol$.

6. Per $tol = 10^{-12}$ e $u_0 = 1/2$ riportare il valore di $u_k$ ed il numero di iterazioni eseguite dal metodo delle tangenti.

**Esercizio 53.** Si consideri l'equazione

$$f(x) = x - \cos x = 0, \quad x \in \mathbb{R}.$$

1. Si dimostri che l'equazione ammette un'unica radice positiva indicata con $\alpha$.

2. Si dimostri che il metodo iterativo $x_{k+1} = g(x_k)$, $g(x) = \cos x$, è localmente convergente in $\alpha$.

3. Si dimostri che il metodo iterativo $x_{k+1} = g(x_k)$, $g(x) = \cos x$, genera successioni convergenti ad $\alpha$ per ogni punto iniziale $x_0 \in [0, \frac{\pi}{2}]$.

4. Scrivere una funzione Matlab che dati in input $x_0 = y_0 \in [0, \frac{\pi}{2}]$ e $tol \in R^+$ genera due successioni $x_{k+1} = g(x_k)$ e $y_{k+1} = y_k - \dfrac{f(y_k)}{f'(y_k)}$, $k > 0$, arrestandosi quando $f_x = |f(x_k)|$ o $f_y = |f(y_k)|$ risulta minore di $tol$. La funzione deve restituire in output i valori di $f_x$, $f_y$ e $k$.

5. Per $tol = 1.0e-8$ e $x_0 = y_0 = 0, \frac{\pi}{4}, \frac{\pi}{2}$, riportare l'output del programma.

**Esercizio 54.** Per calcolare $\sqrt{a}$ con $a > 0$ si considera il metodo delle tangenti applicato per la risoluzione dell'equazione $f(x) = 0$ con $f\colon \mathbb{R}^+ \to \mathbb{R}$, $f(x) = \dfrac{x^2 - a}{\sqrt{2x}}$.

1. Si dimostri che il metodo delle tangenti applicato per la risoluzione dell'equazione $f(x) = 0$ è localmente convergente in $\alpha = \sqrt{a}$.

2. Si valuti $f''(\alpha)$. Cosa si conclude riguardo l'ordine di convergenza locale del metodo?

3. Si dimostri che il metodo delle tangenti genera successioni convergenti per ogni punto iniziale $x_0 \geq \alpha$.

4. Scrivere una funzione MatLab che dati in input $a \in \mathbb{R}^+$, $x_0 \in \mathbb{R}$, $tol \in \mathbb{R}$, e $itmax \in \mathbb{N}$ implementa il metodo delle tangenti applicato all'equazione $f(x) = 0$ con punto iniziale $x_0$. Il metodo si arresta quando $|x_k - x_{k-1}|/|x_k| \leq tol$ o $k \geq itmax$ riportando in output $k$ ed il vettore $[x_1, \ldots x_k]^T$.

5. Per $tol = 1.0e-14$, $x_0 = a = 9$, $itmax = 100$ riportare il valore di $k$ e di $x_k$ calcolati.

6. Con i dati di output calcolare e riportare quindi la sequenza degli errori relativi
$$\xi_j = |x_j - \sqrt{a}|/|\sqrt{a}|, \quad 1 \leq j \leq k.$$

**Esercizio 55.** Sia $f\colon \mathbb{R}^+ \to \mathbb{R}$ definita da $f(x) = x\log(x) + x^2 - 1$. Utilizzando il comando `plot` si riporti il grafico della funzione per $x \in (0, 1]$) mostrando che $f(x)$ ha un punto di minimo locale denotato con $\alpha$. Per determinarne il valore si considera il metodo delle tangenti applicato all'equazione $f'(x) = 0$.

1. Si dica se il metodo è localmente convergente in $\alpha$.

2. Si mostri che il metodo genera successioni convergenti ad $\alpha$ per ogni punto iniziale $x_0 \in (0, \alpha]$.

3. Si mostri che il metodo genera successioni convergenti ad $\alpha$ per ogni punto iniziale $x_0 \in (0, 1)$.

4. Scrivere una funzione Matlab che dati in input $x_0 \in (0,1)$ implementa il metodo delle tangenti applicato all'equazione $f'(x) = 0$ con punto iniziale $x_0$ arrestandosi quando $x_k - x_{k-1} \leq 0$ e $k > 1$ e riportando in output l'approssimazione $x_k$ di $\alpha$ ed il numero $k$ di iterazioni eseguite.

5. Per $x_0 \in \{0.1, 0.5, 0.9\}$ riportare il numero di iterazioni eseguite dal metodo e l'approssimazione determinata.

6. Cosa accade se $x_0 = 1$? Illustrare e giustificare i risultati sperimentali.

**Esercizio 56.** Si consideri il metodo delle tangenti per la risoluzione dell'equazione

$$f(x) = 4x^3 - e^{-x} = 0.$$

1. Si mostri che l'equazione $f(x) = 0$ ammette una sola soluzione reale denotata con $\alpha$. Si mostri che l'equazione $f''(x) = 0$ ammette una sola soluzione reale denotata con $\beta$. Si mostri che vale $0 < \beta < \alpha < 1$.

2. Si dimostri che $\forall\ x_0 > \beta$ la successione generata dal metodo delle tangenti è convergente ad $\alpha$. Si dimostri che $\forall\ x_0 \leq \beta$ la successione generata dal metodo delle tangenti è tale che $\exists\ k > 0$ con $x_k > \beta$.

3. Scrivere una funzione MatLab che dati in input $tol \in \mathbb{R}$ e $x_0$ genera la successione generata dal metodo delle tangenti a partire da $x_0$ arrestandosi quando $|x_k - x_{k-1}| \leq tol$ e restituendo in uscita la coppia $(x_k, k)$. Riportare il numero di iterazioni ottenute per $x_0 = 1$ e $x_0 = -1$ e $tol = 1.0e-8$.

**Esercizio 57.** Si consideri l'equazione

$$f(x) = x - \frac{\pi}{2} - \arctan(x) = 0$$

1. Si mostri che l'equazione ammette una sola soluzione reale denotata con $\alpha$. Si mostri che $\alpha \in [\pi/2, \pi]$.

2. Si consideri il metodo iterativo $x_{k+1} = g(x_k) = \frac{\pi}{2} + \arctan(x_k)$. Si mostri che $\alpha$ è punto fisso di $g(x)$ e $0 < g'(x) = \dfrac{1}{1+x^2} \leq 1$. Si mostri che il metodo iterativo genera successioni convergenti ad $\alpha$ per ogni scelta del punto iniziale in $[\pi/2, \pi]$. Si mostri che il metodo genera successioni convergenti ad $\alpha$ per ogni scelta del punto iniziale.

3. Scrivere una funzione MatLab che dati in input $x_0$ genera la successione generata dal metodo iterativo $x_{k+1} = g(x_k)$ a partire da $x_0$ arrestandosi quando $|x_k - x_{k-1}| \leq 10$ `eps` e restituendo in uscita la coppia $(x_k, k)$. Riportare il valore di $k$ e $x_k$ generato per $x_0 = \pi$.

**Esercizio 58.** Si consideri l'equazione

$$f(x) = x\ e^x - 1 = 0.$$

1. Si mostri che l'equazione ha un'unica radice reale $\alpha$ e $\alpha \in [0, 1]$.

2. Posto $\phi(x) = e^{-x}$ si mostri che la successione

$$\begin{cases} z_0 = 1; \\ z_{k+1} = \phi(z_k), \ k \geq 0; \end{cases}$$

converge ad $\alpha$.

3. Si mostri inoltre che la successione generata dal metodo delle tangenti applicato per la risoluzione dell'equazione $f(x) = 0$ con punto iniziale $x_0 = 1$ risulta convergente ad $\alpha$.

4. Si riporti il valore di $\alpha$ determinato dalla dalla funzione MatLab `fzero` applicata per la risoluzione dell'equazione nell'intervallo $[0, 1]$.

5. Assegnando ad $\alpha$ il valore determinato al punto precedente si riportino su un grafico i valori dei vettori $\boldsymbol{y} = [y_1, \ldots, y_{10}]$ e $\boldsymbol{w} = [w_1, \ldots, w_{10}]$ definiti da

$$y_k = \begin{cases} \dfrac{|z_k - \alpha|}{|z_{k-1} - \alpha|} \text{ se } |z_{k-1} - \alpha| \neq 0; \\ 0 \text{ altrimenti}; \end{cases}$$

e

$$w_k = \begin{cases} \dfrac{|x_k - \alpha|}{|x_{k-1} - \alpha|} \text{ se } |x_{k-1} - \alpha| \neq 0; \\ 0 \text{ altrimenti}. \end{cases}$$

Commentare i risultati.

**Esercizio 59.** Nel 1224 Leonardo Fibonacci fornisce la seguente approssimazione $\xi$ di una soluzione reale dell'equazione

$$f(x) = x^3 + 2x^2 + 10x - 20 = 0,$$

$$\xi = 1 + 22\frac{1}{60} + 7\left(\frac{1}{60}\right)^2 + 42\left(\frac{1}{60}\right)^3 + 33\left(\frac{1}{60}\right)^4 + 4\left(\frac{1}{60}\right)^5 + 40\left(\frac{1}{60}\right)^6.$$

1. Determinare il numero di soluzioni reali dell'equazione.

2. Scrivere una funzione MatLab che dati in input $x_0, tol$ implementa il metodo delle tangenti per l'approssimazione della soluzione reale dell'equazione $f(x) = 0$ con punto iniziale $x_0$ arrestandosi quando $|x_{k+1} - x_k| \leq tol$.

3. Determinare un punto iniziale $x_0$ tale da garantire (teoricamente) la convergenza.

4. Valutare l'accuratezza dell'approssimazione fornita da Fibonacci.

**Esercizio 60.** Si consideri l'equazione $t^3 - 2t - 5 = 0$.

1. Si determini il numero di soluzioni reali dell'equazione.

2. Utilizzando il metodo di bisezione si determina un'approssimazione $\eta$ della radice reale positiva, indicata con $\chi$, tale da aversi $|\eta - \chi| \leq 2^{-5}$. Dire quanti passi del metodo sono sufficienti a garantire l'accuratezza richiesta.

3. Considerati i seguenti metodi iterativi

$$t_{i+1} = t_i^3 - t_i - 5, \quad t_{i+1} = t_i - (t_i^3 - 2t_i - 5)/10,$$

si dica quale risulta localmente convergente in un intorno di $\chi$.

**Esercizio 61.** Si consideri la questione descritta nel seguente documento.

# Chapter 4. Optimization

## 4.1 Introduction

Optimization problems are ubiquitous in science and engineering, and even in our daily life, thinking about how we optimize our way to go to work, the choice of line we stand in at the supermarket, or deciding for the education for our children.

We begin this chapter with several simple examples, which show the breadth of problems that fall into the category of optimization problems.

### 4.1.1 How much daily exercise is optimal ?

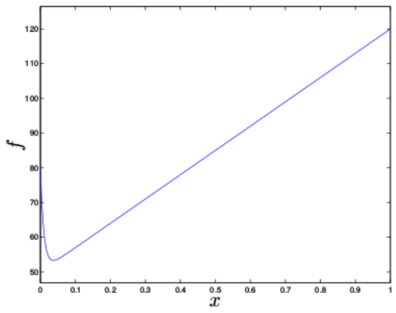
In his John von Neumann lecture at the annual SIAM meeting, Joe Keller asked this question and proposed a very simple model to answer it. Suppose at birth, every human being is given a fixed number of heart beats, and once these heartbeats are used up, life ends. How should one optimally use these heart beats to have as long a life as possible ? A first immediate idea is to stay in bed and rest, so the heart beat stays low, and one uses the heart beats as economically as possible. Another idea comes however from the fact that a well trained heart is beating much more slowly when the person is at rest, than the heart of an untrained person. So exercise could be beneficial to increase the lifespan. Unfortunately during exercise, the heart beats fast, so that one uses up heart beats faster, in the hope to gain them back during rest. So is there an optimum ?

Suppose that the untrained heart beats 80 times a minute when a person is at rest, and that during exercise, it beats 120 times per minute. If a person exercises the fraction $x$ of its time, then this person uses on average

$$f(x) := 120x + g(x)(1 - x)$$

heartbeats per minute, where the unknown function $g$ should be close to 80 for $x$ small, which means the person does hardly do any exercise, and probably around 50 for $x$ approaching 1, which means that the person is extremely well trained. Since it is known that a little exercise every day decreases the heart beat at rest already considerably, a simple model for $g$ would be exponential decay, i.e.

$$g(x) := 50 + 30e^{-100x},$$




where the choice $-100$ is quite arbitrary here, and should be much more carefully researched with the help of a medical doctor. Figure 4.1.1 shows the function $f$ for this example, and there is clearly a minimum, which means there is an optimum choice of $x$, which minimizes the average use of heartbeats, and thus leads to the longest life possible. From calculus, we know that we need to set the derivative to zero to find the minimum, which with Maple is easily achieved by

```
f:=120*x+(50+30*exp(-100*x))*(1-x);
fp:=diff(f,x);
solve(fp,x);
```

$$-\frac{1}{100}\,\mathrm{LambertW}(\frac{7}{3}\,e^{101})+\frac{101}{100}$$

It is interesting to see that the closed form solution of this problem involves the LambertW function we have already encountered in Chapter **??** on the numerical solution of non-linear equations. To obtain a numerical value in Maple, we type

```
evalf(%);
```

and Maple returns 0.0373019079, which means that one should exercise a bit over 50 minutes every day. If there had been no closed form solution in Maple, one could have used any of the nonlinear equation solvers from Chapter **??** to find the solution, or the maple command `fsolve`.

Is it also possible to find the minimum directly without knowing the derivative ? For this one dimensional problem, an algorithm like bisection from Section **??** would be nice, but it is not possible to tell from the midpoint of an initial interval $[a, b]$ if the minimum lies in the left or right half of the interval. If one however computes the function value for two distinct points $x_1$ and $x_2$ in $[a, b]$, $x_1 < x_2$, and if $f(x_1)$ is smaller than $f(x_2)$, as in the example in Figure 4.1, then a minimum must lie in the interval $[a, x_2]$. On the other hand, if $f(x_1)$ is bigger than $f(x_2)$, then a minimum must lie in the interval $[x_1, b]$, and hence

1. Mediante il comando `plot` tracciare il grafico della funzione `f` $= f(x)$ per $0 \leq x \leq 1$.

2. Determinare il numero di soluzioni reali dell'equazione $f'(x) = 0$ nell'intervallo $(0, 1)$.

3. Implementare il metodo di bisezione ed approssimare le soluzioni dell'equazioni con errore assoluto minore di $2^{-34}$.

4. Confrontare l'approssimazione delle soluzioni con la forma esplicita descritta nel documento (per il calcolo della funzione `LambertW` si può utilizzare la funzione `lambertw` implementata nel pacchetto simbolico di MatLab).

# Capitolo 11

# Interpolazione Polinomiale ed Integrazione Numerica

## Lezione 11.1: Il Problema dell'Interpolazione Polinomiale.

L'approssimazione mediante polinomi riveste una notevole importanza data la relativa facilità con cui quest'ultimi possono essere manipolati in un ambiente computazionale. Le tecniche di approssimazione basate sul processo di *interpolazione* poggiano sul seguente risultato di *esistenza ed unicità*.

**Teorema 11.1.1.** Sia $\Pi_n$ lo spazio vettoriale dei polinomi a coefficienti reali di grado minore od uguale ad $n$ e sia $\Phi = \{\phi_0(x), \ldots, \phi_n(x)\}$ una base di $\Pi_n$. Assegnate $(x_i, y_i) \in \mathbb{R}^2$, $0 \leq i \leq n$, $n+1$ coppie di numeri reali con $x_i \neq x_j$ se $i \neq j$, esiste ed è unico il polinomio $p(x) \in \Pi_n$, $p(x) = \sum_{i=0}^{n} \alpha_i \phi_i(x)$ tale che

$$p(x_k) = \sum_{i=0}^{n} \alpha_i \phi_i(x_k) = y_k, \quad 0 \leq k \leq n. \tag{11.1}$$

Tale polinomio è detto *il polinomio di interpolazione* sui punti $(x_i, y_i) \in \mathbb{R}^2$, $0 \leq i \leq n$.

*Dimostrazione.* La dimostrazione segue osservando che le condizioni di interpolazione (11.1) definiscono un sistema lineare

$$\begin{bmatrix} \phi_0(x_0) & \ldots & \ldots & \phi_n(x_0) \\ \phi_0(x_1) & \ldots & \ldots & \phi_n(x_1) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_0(x_n) & \ldots & \ldots & \phi_n(x_n) \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix},$$

e dunque l'esistenza e l'unicità del polinomio di interpolazione sui punti $\mathcal{S} = \{(x_i, y_i) \in \mathbb{R}^2, 0 \leq i \leq n\}$, segue se mostriamo che la matrice dei coefficienti è invertibile. A tal fine detta $V(\mathcal{S}, \Phi)$ tale matrice proviamo che $\mathrm{Ker}(V(\mathcal{S}, \Phi) = \{\mathbf{0}\}$. Sia $\boldsymbol{\beta} = [\beta_0, \ldots, \beta_n]^T$ un vettore del nucleo. Dalla relazione

$$V(\mathcal{S}, \Phi)\boldsymbol{\beta} = \mathbf{0},$$

segue che il polinomio $b(x) = \sum_{i=0}^{n} \beta_i \phi_i(x)$ di grado al più $n$ si annulla in almeno $n+1$ punti distinti e pertanto per il principio di identità dei polinomi $b(x)$ è identicamente nullo. Essendo $\Phi$ una base di $\Pi_n$ ne discende che $\beta_0 = \beta_1 = \ldots = \beta_n = 0$. □

Se l'insieme dei punti $\mathcal{S} = \{(x_i, y_i) \in \mathbb{R}^2, 0 \leq i \leq n\}$ soddisfa $x_i \neq x_j$ per $1 \leq i \neq j \leq n$, allora il teorema afferma l'esistenza e l'unicità del polinomio di interpolazione $p(x)$. La rappresentazione ed il calcolo dei coefficienti della rappresentazione dipendono dalla scelta della base $\Phi$ di $\Pi_n$.

1. Se $\phi_j(x) = x^j$, $0 \leq j \leq n$, è l'usuale base dei *monomi* allora

$$V(\mathcal{S}, \Phi) = \begin{bmatrix} 1 & \ldots & \ldots & x_0^n \\ 1 & \ldots & \ldots & x_1^n \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \ldots & \ldots & x_n^n \end{bmatrix},$$

   è detta *matrice di Vandermonde*. Il calcolo dei coefficienti del polinomio di interpolazione nella base dei monomi risulta generalmente *mal condizionato*. La risoluzione del sistema lineare richiede al più $O(n^3)$ operazioni aritmetiche.

2. Se $\phi_0(x) = 1$, $\phi_j(x) = \prod_{i=0}^{j-1}(x - x_i)$, $1 \leq j \leq n$, allora matrice associata $V(\mathcal{S}, \Phi)$ risulta triangolare inferiore. Infatti vale

$$(V(\mathcal{S}, \Phi))_{h+1,k+1} = \prod_{i=0}^{k-1}(x_h - x_i) = 0 \text{ se } h < k.$$

   La rappresentazione di $p(x)$ è detta *forma di Newton* del polinomio di interpolazione. Il calcolo dei coefficienti del polinomio di interpolazione nella base di Newton risulta generalmente *mal condizionato*. La risoluzione del sistema lineare richiede al più $O(n^2)$ operazioni aritmetiche.

3. Se

$$\phi_j(x) = L_j(x) = \prod_{i=0, i\neq j}^{n} \frac{x - x_i}{x_j - x_i}, \quad 0 \leq j \leq n,$$

   allora $V(\mathcal{S}, \Phi) = I_n$. Infatti vale

$$(V(\mathcal{S}, \Phi))_{h+1,k+1} = L_k(x_h) = \begin{cases} 1 \text{ se } h = k; \\ 0 \text{ altrimenti.} \end{cases}$$

   La rappresentazione di $p(x) = \sum_{j=0}^{n} y_j L_j(x)$ è detta *forma di Lagrange* del polinomio di interpolazione. Il calcolo dei coefficenti $\alpha_j = y_j$ risulta ottimamente condizionato e non richiede operazioni aritmetiche.

In molteplici contesti applicativi piuttosto che ai coefficenti della rappresentazione si è interessati alla valutazione del polinomio di interpolazione in punti differenti dai nodi. Per la forma di Lagrange si ottiene

$$p(\hat{x}) = \sum_{j=0}^{n} y_j L_j(\hat{x}) = \sum_{j=0}^{n} y_j \prod_{i=0, i\neq j}^{n} \frac{\hat{x} - x_i}{x_j - x_i} = \prod_{i=0}^{n}(\hat{x} - x_i) \sum_{j=0}^{n} \frac{y_j}{\omega_j(\hat{x} - x_j)},$$

con $\omega_j = \prod_{i=0, i\neq j}^{n}(x_j - x_i)$, $0 \leq j \leq n$. Se allora i pesi $\omega_j$ sono precomputati e già disponibili la valutazione di $p(\hat{x})$ richiede al più $O(n)$ operazioni aritmetiche.

# Lezione 11.2: Resto dell' Interpolazione Polinomiale.

Se $y_j = f(x_j)$, $0 \leq j \leq n$, allora le tecniche di interpolazione consentono di approssimare la funzione eventualmente incognita $f$ in punti differenti dai nodi. In tal caso risulta essenziale disporre di misure dell'errore commesso nell'approssimazione. Il seguente è detto *teorema del resto dell'interpolazione polinomiale*.

**Teorema 11.2.1.** Sia $f\colon [a,b] \to \mathbb{R}$, $f \in C^{n+1}([a,b])$, e siano $a \leq x_0 < x_1 < \ldots < x_n \leq b$ $n+1$ nodi distinti. Detto $p(x)$ il polinomio di interpolazione sui punti $\mathcal{S} = \{(x_i, f(x_i)) \in \mathbb{R}^2, 0 \leq i \leq n\}$ si ha

$$\forall \hat{x} \in [a,b]\ \exists \hat{\xi} \in [a,b]\colon f(\hat{x}) - p(\hat{x}) = \frac{f^{(n+1)}(\hat{\xi})}{(n+1)!} \prod_{i=0}^{n} (\hat{x} - x_i).$$

*Dimostrazione.* Se $\hat{x} = x_i$ per un certo $i$ allora $f(\hat{x}) - p(\hat{x}) = f(x_i) - p(x_i) = 0$ per le condizioni di interpolazione ed inoltre $\prod_{j=0}^{n}(\hat{x} - x_j) = \prod_{j=0}^{n}(x_i - x_i) = 0$. Segue dunque che la relazione è verificata per ogni $\hat{\xi} \in [a,b]$.

Si supponga ora $\hat{x} \neq x_i$, $0 \leq i \leq n$. Si consideri la funzione ausiliaria

$$h(x) = f(x) - p(x) - \frac{f(\hat{x}) - p(\hat{x})}{\prod_{j=0}^{n}(\hat{x} - x_j)} \prod_{i=0}^{n} (x - x_i).$$

Si osservi che $h(x_i) = 0$, $0 \leq i \leq n$, e $h(\hat{x}) = 0$. Pertanto la funzione si annulla in almeno $n+2$ punti distinti in $[a,b]$. Inoltre $h(x)$ eredita la regolarità di $f(x)$ essendo le altre due componenti polinomi in $x$ di grado rispettivamente $\leq n$ e $n+1$. Dal teorema di Rolle segue che $h'(x)$ si annulla in almeno $n$ punti distinti in $[a,b]$ ed, iterando il ragionamento, che $h^{(n+1)}(x)$ si annulla in almeno 1 punto detto $\hat{\xi} \in [a,b]$. Si ha pertanto

$$0 = h^{(n+1)}(\hat{\xi}) = f^{(n+1)}(\hat{\xi}) - \frac{f(\hat{x}) - p(\hat{x})}{\prod_{j=0}^{n}(\hat{x} - x_j)}(n+1)!,$$

da cui la tesi. □

Se l'intervallo $[a,b]$ è sufficientemente piccolo o se $f^{(n+1)}(x)$ non varia molto in $[a,b]$ allora ne segue che la qualità dell'approssimazione dell'interpolante polinomiale dipende essenzialmente dal fattore $\prod_{i=0}^{n}(\hat{x} - x_i)$. Se $x_j = a + j\frac{b-a}{n}$, $0 \leq j \leq n$, sono equispaziati in $[a,b]$ allora dal grafico di $\omega(x) = \prod_{i=0}^{n}(x - x_i)$ si evince che la qualità è migliore per punti $\hat{x}$ centrali mentre si deteriora procedendo verso gli estremi dell'intervallo. In particolare l'andamento oscillante del grafico e l'aumentare della ampiezza delle oscillazioni agli estremi dell'intervallo suggerisce la possibilità di errori di approssimazione elevati in prossimità degli estremi dell'intervallo. Tale evidenza sperimentale è confermata teoricamente come mostra il seguente *esempio di Runge*. Sia $f(x) = 1/(1+x^2)$ $a = -5, b = 5$, $x_j = -5 + j(10/n)$, $0 \leq j \leq n$. Detto $p_n(x)$ il polinomio di interpolazione sui nodi $\mathcal{S} = \{(x_i, f(x_i)) \in \mathbb{R}^2, 0 \leq i \leq n\}$ si ha che $\exists \gamma \in (3,4)$ tale che

$$\lim_{n \to +\infty} |f(\hat{x}) - p(\hat{x})| = +\infty, \quad \forall \hat{x}\colon \gamma \leq |\hat{x}| \leq 5.$$

# Lezione 11.3: Integrazione Numerica.

Una delle applicazioni più interessanti dei metodi di interpolazione polinomiale concerne la sintesi di algoritmi numerici per l'approssimazione dell'integrale definito $\mathcal{I}(f,a,b) = \int_a^b f(x)dx$. L'approccio seguente conduce alle *formule di Newton-Cotes*. Posto $x_j = a + j\frac{b-a}{n}$, $0 \le j \le n$, $n+1$ punti equispaziati in $[a,b]$ e detto $p_n(x)$ il polinomio di interpolazione sui nodi $\mathcal{S} = \{(x_i, f(x_i)) \in \mathbb{R}^2, 0 \le i \le n\}$ si considera l'approssimazione di $\mathcal{I}(f,a,b)$ fornita da $\mathcal{I}(p_n,a,b)$. Si ha

$$\mathcal{I}(p_n,a,b) = \int_a^b p_n(x)dx = \int_a^b \sum_{j=0}^n y_j L_j(x)dx = \sum_{j=0}^n y_j \int_a^b L_j(x)dx.$$

Inoltre posto $x = a + t\frac{b-a}{n}$, $t \in \mathbb{R}$, $0 \le t \le n$, vale per $0 \le j \le n$,

$$\int_a^b L_j(x)dx = \int_a^b \prod_{i=0, i\neq j}^n \frac{x-x_i}{x_j-x_i}dx = \frac{b-a}{n}\int_0^n \prod_{i=0, i\neq j}^n \frac{t-i}{j-i}dt = \frac{b-a}{n}\sigma_j^{(n)},$$

dove i pesi $\sigma_j^{(n)}$, $0 \le j \le n$, sono funzioni di $j$ ed $n$ e non dipendono né dalla funzione integranda $f$ e né dall'intervallo di integrazione $[a,b]$ e pertanto possono essere precomputati e tabulati. L'approssimazione così ottenuta

$$\mathcal{I}(p_n,a,b) = \frac{b-a}{n}\sum_{j=0}^n y_j\sigma_j^{(n)} = \frac{b-a}{n}\sum_{j=0}^n f(x_j)\sigma_j^{(n)},$$

è detta *formula di Newton-Cotes* su $n+1$ nodi. Per $n = 1$ si ottiene la *formula dei trapezi*. Per $n = 2$ si ottiene la *formula di Cavalieri-Simpson*. Per $n \ge 7$ compaiono pesi negativi il che comporta difficoltà numeriche e preclude teoricamente ad una dimostrazione della convergenza di $\mathcal{I}(p_n,a,b)$ a $\mathcal{I}(f,a,b)$ –risultato peraltro atteso se pensiamo al comportamento dell'interpolazione polinomiale su nodi equidistanti (esempio di Runge)–.

Per ottenere una sequenza di approssimazioni convergenti si procede come segue. Si ha

$$\mathcal{I}(f,a,b) = \int_a^b f(x)dx = \sum_{j=0}^{n-1}\mathcal{I}(f,x_j,x_{j+1}).$$

Per l'approssimazione di ciascun integrale $\mathcal{I}(f,x_j,x_{j+1})$, $0 \le j \le n-1$, si utilizza una formula di Newton-Cotes su $m+1$ punti con $m$ fissato e $m \le 6$. Si parla in tal caso di applicazione iterata della formula in oggetto. Per $m = 1$ si ha

$$\mathcal{I}_1^{(n)} = \sum_{j=0}^{n-1}\mathcal{I}(p_1,x_j,x_{j+1}) = \frac{b-a}{n}\sum_{j=0}^{n-1}\frac{f(x_j)+f(x_{j+1})}{2}.$$

detta *formula dei trapezi iterata*. Per una stima dell'errore commesso nell'approssimazione di $\mathcal{I}(f,a,b)$ con $\mathcal{I}_1^{(n)}$ si assuma che $f \in C^2([a,b])$. Si ponga

$$\mathcal{E}_1^{(n)} = \mathcal{I}(f,a,b) - \mathcal{I}_1^{(n)} = \sum_{j=0}^{n-1}(\mathcal{I}(f,x_j,x_{j+1}) - \mathcal{I}(p_1,x_j,x_{j+1})).$$

Dal teorema del resto dell'interpolazione polinomiale segue che

$$|\mathcal{I}(f,x_j,x_{j+1})-\mathcal{I}(p_1,x_j,x_{j+1})| \leq \frac{M}{2}\int_{x_j}^{x_{j+1}}(x-x_j)(x_{j+1}-x)dx, \quad 0 \leq j \leq n-1$$

con $M = \max_{x\in[a,b]} |f''(x)|$ e quindi

$$|\mathcal{I}(f,x_j,x_{j+1}) - \mathcal{I}(p_1,x_j,x_{j+1})| \leq \frac{M}{12}\frac{(b-a)^3}{n^3}, \quad 0 \leq j \leq n-1.$$

Ne discende che

$$0 \leq |\mathcal{E}_1^{(n)}| \leq n\frac{M}{12}\frac{(b-a)^3}{n^3} = \frac{M}{12}\frac{(b-a)^3}{n^2},$$

e pertanto dal teorema del confronto

$$\lim_{n\to+\infty} |\mathcal{E}_1^{(n)}| = \lim_{n\to+\infty} |\mathcal{I}(f,a,b) - \mathcal{I}_1^{(n)}| = 0.$$

Risultati analoghi si trovano per le altre formule iterate. In pratica se è disponibile una stima (maggiorazione) di $M$ il precedente risultato ci descrive una strategia per la determinazione di $n$ in modo da garantire una fissata accuratezza nell'approssimazione dell'integrale. Infatti detta *tol* questa accuratezza si ha che se il valore di $n$ è determinato in modo tale che

$$\frac{M}{12}\frac{(b-a)^3}{n^2} \leq tol$$

allora si ha

$$|\mathcal{I}(f,a,b) - \mathcal{I}_1^{(n)}| \leq tol.$$

Se tale stima non è disponibile si può applicare una strategia dove ad ogni passo si raddoppia il numero dei punti fintanto che due stime consecutive non sono sufficientemente vicine. L'assunzione di nodi equidistanti è una semplificazione forte. In realtà nelle applicazioni pratiche si usano distribuzioni di nodi *adattative* in cui i nodi vengono addensati in modo automatico dove necessario (es. in zone con brusche variazioni della funzione integranda). Ad esempio si possono raddoppiare i nodi nei soli intervalli che presentano scostamenti significativi.

# Lezione 11.4: Esercizi.

**Esercizio 62.** La concentrazione di una certa tossina in un sistema di laghi situati in un'area industriale è stato monitorata ad intervalli di tempo nel periodo 1978–1992 come mostrato nella tabella riportata di seguito:

| year | toxin |
|---|---|
| 1978 | 12 |
| 1980 | 12.7 |
| 1982 | 13 |
| 1984 | 15.2 |
| 1986 | 18.2 |
| 1988 | 19.8 |
| 1990 | 24.1 |
| 1992 | 28.1 |

1. Interpolare i dati utilizzando la forma di Lagrange del polinomio di interpolazione. Tracciare il grafico del polinomio mediante il comando `plot`. Utilizzare il polinomio per predire il valore della tossina nell'anno 1994.

2. L'interpolazione polinomiale potrebbe essere usata anche per riempire dei "buchi" nei dati. Supponiamo di non disporre delle misurazioni per gli anni 1982 e 1984. Determinare il polinomio di interpolazione sui dati rimanenti e calcolare le predizioni mancanti.

3. Per risolvere il punto precedente considerare il seguente approccio alternativo. Determinare una funzione polinomiale a tratti $s(t)$ definita come
$$s(t) = \begin{cases} s_1(t) \text{ per } t \in [t_0, t_1] = [1978, 1980]; \\ s_2(t) \text{ per } t \in [t_1, t_2] = [1980, 1986]; \\ s_3(t) \text{ per } t \in [t_2, t_3] = [1986, 1988]; \\ s_4(t) \text{ per } t \in [t_3, t_4] = [1988, 1990]; \\ s_5(t) \text{ per } t \in [t_4, t_5] = [1990, 1992]; \end{cases}$$
con $s_i(t) = a_i + b_i t + c_i t^2 + d_i t^3$, $1 \le i \le 5$, e i coefficienti $a_i, b_i, c_i, d_i$ determinati imponendo le condizioni di interpolazione $s_i(t_{i-1}) = y_{i-1}$ e $s_i(t_i) = y_i$ con $y_i$ il valore della tossina al tempo $t_i$, $1 \le i \le 5$, le condizioni di raccordo $s_i'(t_i) = s_{i+1}'(t_i)$ e $s_i''(t_i) = s_{i+1}''(t_i)$, $1 \le i \le 4$ e le due condizioni aggiuntive $s_1''(t_0) = s_5''(t_5) = 0$. Risolvere il sistema lineare nelle incognite $a_i, b_i, c_i, d_i$, $1 \le i \le 5$. Tracciare un grafico di $s(x)$ e calcolare $s(1982)$ e $s(1984)$.

**Esercizio 63.** Dati $x_k = k/n$, $1 \le k \le n$, $n \ge 1$, sia $A_n = (a_{i,j}) \in \mathbb{R}^{n \times n}$ la matrice definita da $a_{i,j} = x_j^{i-1}$, $1 \le i, j \le n$. Per $n = 4$ si ottiene
$$A_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1/4 & 2/4 & 3/4 & 4/4 \\ (1/4)^2 & (2/4)^2 & (3/4)^2 & (4/4)^2 \\ (1/4)^3 & (2/4)^3 & (3/4)^3 & (4/4)^3 \end{bmatrix}.$$

1. Si mostri che $A_n$ è invertibile.

2. Posto
$$L_j(x) = \prod_{i=1, i \ne j}^{n} \frac{x - x_i}{x_j - x_i} = \sum_{i=0}^{n-1} p_i^{(j)} x^i, \quad 1 \le j \le n,$$
il $j-$esimo polinomio di Lagrange e la sua rappresentazione nella usuale base dei monomi, si mostri che
$$\left[ p_0^{(j)}, p_1^{(j)}, \ldots, p_{n-1}^{(j)} \right] A_n = e_j^T, \quad 1 \le j \le n,$$
con $e_j$ il $j-$esimo vettore della base canonica.

3. Per $n$ dispari si determini $L_{(n+1)/2}(0) = p_0^{(n+1)/2}$ e se ne deduca una limitazione inferiore per il condizionamento in norma infinito di $A_n$.

4. Scrivere una funzione Matlab che dati in input $n \in \mathbb{N}$ restituisce in output il vettore $[L_1(0), L_2(0), \ldots L_n(0)]$ formato dalle valutazioni dei polinomi di Lagrange nell'origine.

5. Si determini il costo computazionale dell'algoritmo implementato.

6. Per $n = 41$ si riporti il plot del vettore restituito dall'implementazione.

**Esercizio 64.** Si consideri il calcolo dell'integrale

$$I_s = \int_0^1 \frac{\sin x}{\sqrt{x}}\, \mathrm{d}x.$$

1. Scrivere una funzione Matlab® che dato in input il naturale $n$ restituisce in output l'approssimazione dell'integrale fornita dalla formula dei trapezi composta su $n$ intervalli applicata per il calcolo dell'integrale assumendo di estendere per continuità la funzione integranda nell'origine.

2. Effettuare il cambio di variabile $x = t^2$ e scrivere una funzione Matlab® che dato in input il naturale $n$ restituisce in output l'approssimazione dell'integrale fornita dalla formula dei trapezi composta su $n$ intervalli applicata per il calcolo dell'integrale modificato.

3. Assumendo $I_s = 0.62053660344676$, confrontare gli errori generati nei due approcci descritti sopra e spiegarne teoricamente il comportamento.

**Esercizio 65.** Siano assegnate le coppie di punti $(x_i, f_i) \in \mathbb{R}^2$, $0 \le i \le n$, con $x_i \neq x_j$ se $i \neq j$ e $f_i = f(x_i)$, $f\colon \mathbb{R} \to \mathbb{R}$.

1. Determinare $\gamma_0, \ldots, \gamma_n$ in modo tale che la funzione

$$r(x) = \frac{\sum_{i=0}^n \dfrac{(-1)^i \gamma_i}{x - x_i}}{\sum_{i=0}^n \dfrac{(-1)^i}{x - x_i}},$$

soddisfi le condizioni di interpolazione

$$r(x_i) = f(x_i) = f_i, \quad 0 \le i \le n.$$

2. Posto $f(x) = 1/(1 + x^2)$ e $x_i = -5 + \dfrac{10}{n} i$, $0 \le i \le n$, utilizzando Matlab si tracci un grafico approssimativo della funzion razionale $r(x)$ per $n = 32, 64, 128$. Cosa si conclude circa la localizzazione dei poli della funzione razionale?

3. Si valuti inoltre per le tre funzioni razionali generate al variare di $n$ l'errore di approssimazione

$$\epsilon_n = \max_i |f(x_i) - r(x_i)|,$$

con $x_i = -5 + \dfrac{10}{511} i$, $1 \le i \le 510$.

4. Cosa suggerisce questa valutazione rispetto alla convergenza delle approssimanti razionali così costruite alla funzione $f(x)$ nell'intervallo $[-5, 5]$?

**Esercizio 66.** Sia $f(x) = \dfrac{\pi - 2\cos(\pi x)}{2\pi}$, $0 \le x \le 1$.

1. Si calcoli $I = \int_0^1 f(x)\mathrm{d}x$.
2. Si determini l'appprossimazione $I(n)$ di $I$ ottenuta valutando l'integrale con la formula dei trapezi composta su $n$ sottointervalli.
3. Si mostri che la formula restituisce il valore esatto dell'integrale.
4. Si verifichi che lo stesso risultato vale per un'arbitraria funzione continua sull'intervallo $[0, 1]$ che verifica

$$f(x) + f(1 - x) = 1, \quad 0 \leq x \leq 1.$$

**Esercizio 67.** Si intende approssimare la funzione $f(t) = \sin\left(\frac{\pi}{2}t\right)$ per $0 \leq t \leq 1$ con un polinomio $p_n(t)$ della forma

$$p_n(t) = t + t(1 - t)\sum_{j=1}^{n} c_j t^{j-1}. \tag{11.2}$$

Posto $t_k = \dfrac{k}{n+1}$, $1 \leq k \leq n$, si consideri il seguente sistema lineare determinato dalle condizioni di interpolazione

$$\begin{cases} p_n(t_1) = f(t_1) \\ p_n(t_2) = f(t_2) \\ \vdots \quad \vdots \quad \vdots \\ p_n(t_n) = f(t_n) \end{cases} \tag{11.3}$$

nelle incognite $c_1, \ldots, c_n$.

1. Dimostrare che esiste ed è unica la soluzione del sistema lineare (11.5).
2. Scrivere una funzione MatLab che dati in input il valore di $n \in \mathbb{N}$ utilizzando l'operatore "backslash" \ restituisce in output la soluzione $[c_1, \ldots, c_n]^T$ del sistema lineare (11.5).
3. Per $n = 5$ e $n = 15$ riportare il numero di condizionamento in norma infinito della matrice dei coefficienti del sistema lineare valutato dalla funzione `cond`.
4. Posto $\boldsymbol{y}^T = [y_1, \ldots, y_{1000}]$ con $y_i = \dfrac{i-1}{999}$, $1 \leq i \leq 1000$, calcolare

$$e_5 = \max_i |f(y_i) - p_5(y_i)|, \quad e_{15} = \max_i |f(y_i) - p_{15}(y_i)|,$$

   dove $p_5(t)$ e $p_{15}(t)$ sono determinati come in (11.4) a partire dai coefficienti calcolati dalla funzione rispettivamente per $n = 5$ e $n = 15$.
5. Dimostrare che $p_n(t)$ è il polinomio di interpolazione alla funzione $f(t)$ sui nodi $0, t_1, \ldots, t_n, 1$.
6. Mediante il teorema del resto dell'interpolazione polinomiale determinare quindi una maggiorazione dell'errore

$$\epsilon_n = \max_{0 \leq t \leq 1} |f(t) - p_n(t)|.$$

**Esercizio 68.** Data $f(x) \in C^2[a,b]$ si considerino le approssimazioni $I_R(n)$ e $I_T(n)$ del il valore dell'integrale definito $I = \int_a^b f(x)dx$ ottenute rispettivamente mediante la formula composta dei rettangoli e dei trapezi su $n$ sottointervalli,

$$I_R(n) = \frac{b-a}{n}\sum_{k=0}^{n-1} f(x_k), \quad I_T(n) = \frac{b-a}{n}\sum_{k=0}^{n-1} \frac{f(x_k)+f(x_{k+1})}{2}$$

con $x_k = a + k\dfrac{b-a}{n}$, $0 \le k \le n$.

1. Scrivere una funzione MatLab che dato in input $n \in \mathbb{N}$ restituisce in output le approssimazioni $I_R(n)$ e $I_T(n)$ del valore dell'integrale definito

$$I = \frac{1}{2\pi}\int_0^{2\pi} \sqrt{1 - 0.36\ sin^2\theta}\ d\theta.$$

2. Nel caso considerato con $f(x) = \sqrt{1 - 0.36\ sin^2 x}$ e $[a,b] = [0, 2\pi]$ si dimostri che

$$\forall n \in \mathbb{N} \quad I_R(n) = I_T(n).$$

3. Assunto $I = 0.9027799277721938$ come valore corretto alla precisione di macchina si riportino gli errori $E_R(n) = |I_R(n) - I|$ e $E_T(n) = |I_T(n) - I|$ per $n = 4, 8, 12, 16$.

4. Utilizzando la formula del resto e noto che $\max_{x\in[0,2\pi]} |f''(x)| \le 1$ si determini un valore di $n$ sufficiente a garantire

$$E_T(n) < 10^{-10}.$$

5. Determinare sperimentalmente il minimo $n$ tale per cui

$$E_T(n) < 10^{-10}.$$

**Esercizio 69.** Si intende approssimare la funzione $f(t) = cos(\frac{\pi}{2}t)$ per $-1 \le t \le 1$ con un polinomio $p_n(t)$ della forma

$$p_n(t) = \sum_{j=0}^{n} c_j t^{2j}. \tag{11.4}$$

Posto $t_k = \dfrac{k}{n}$, $0 \le k \le n$, si consideri il seguente sistema lineare determinato dalle condizioni di interpolazione

$$\begin{cases} p_n(t_0) = f(t_0) \\ p_n(t_1) = f(t_1) \\ \vdots \quad \vdots \quad \vdots \\ p_n(t_n) = f(t_n) \end{cases} \tag{11.5}$$

nelle incognite $c_0, \ldots, c_n$.

1. Dimostrare che esiste ed è unica la soluzione del sistema lineare (11.5).

2. Scrivere una funzione MatLab che dati in input il valore di $n \in \mathbb{N}$ utilizzando l'operatore "backslash" \ restituisce in output la soluzione $[c_0, \ldots, c_n]^T$ del sistema lineare (11.5).

3. Per $n = 8$ e $n = 16$ riportare il numero di condizionamento in norma infinito della matrice dei coefficienti del sistema lineare valutato dalla funzione `cond`.

4. Posto $\boldsymbol{y}^T = [y_1, \ldots, y_{1000}]$ con $y_i = -1 + 2\dfrac{i-1}{999}$, $1 \leq i \leq 1000$, calcolare

$$e_8 = \max_i |f(y_i) - p_8(y_i)|, \quad e_{16} = \max_i |f(y_i) - p_{16}(y_i)|,$$

dove $p_8(t)$ e $p_{16}(t)$ sono determinati come in (11.4) a partire dai coefficienti calcolati dalla funzione rispettivamente per $n = 8$ e $n = 16$.

**Esercizio 70.** Nell'analisi statistica dei dati interviene frequentemente il calcolo dell'integrale

$$I(m) = \frac{1}{\sqrt{2\pi}} \int_{-m}^{m} e^{-t^2/2} dt, \quad m > 0.$$

1. Per il calcolo di $I(6)$ con la formula dei trapezi iterata determinare il numero di sottointervalli in cui suddividere l'intervallo di integrazione in modo da ottenere una approssimazione $\hat{I}_6$ tale da aversi

$$|\hat{I}_6 - I(6)| \leq 1.0e - 8.$$

2. Si implementi una funzione MatLab che dato in input il numero $N$ di sottointervalli restituisce in output il valore approssimato dalla formula dei trapezi iterata su $N$ sottointervalli per l'approssimazione di $I(6)$.

3. Posto $N_0 = 12$ e assunto $\hat{I}_6$ come valore esatto di $I(6)$ si determini

$$\epsilon_j = |I_j - \hat{I}_6|, \quad 1 \leq j \leq 6,$$

dove $I_j$ è il valore restituito dalla funzione con input $N_j = N_0 2^{j-1}$.

4. Riportare le quantità $r_i = \epsilon_{i+1}/\epsilon_i$, $1 \leq i \leq 5$. Cosa si osserva circa la riduzione dell'errore?

**Esercizio 71.** Sia $p(x)$ il polinomio di interpolazione alla funzione $f(x) = e^x$ sui nodi $x_0 = 0$, $x_1 = 1/2$ e $x_2 = 1$.

1. Si determini una maggiorazione dell'errore $\max_{0 \leq x \leq 1} |f(x) - p(x)|$.

2. Sia $t(x) = 1 + x + x^2/2$ il polinomio di approssimazione di Taylor. Si determini una maggiorazione dell'errore $\max_{0 \leq x \leq 1} |f(x) - t(x)|$.

3. Si scriva una function MatLab che dati in input i nodi $x_i$, $1 \leq i \leq n$, restituisce in ouput i coefficienti del polinomio di interpolazione di grado al più $n - 1$ alla funzione $f(x) = e^x$ risolvendo il sistema lineare associato mediante l'operatore di backslash.

**Esercizio 72.** Si consideri l'arco parabolico $\Gamma = \{(x, y) \in \mathbb{R}^2 \colon y = \dfrac{x(30-x)}{10},\ 0 \leq x \leq 30\}$. La lunghezza dell'arco è

$$L = \int_0^{30} \sqrt{1 + \left(\frac{\mathrm{d}y}{\mathrm{d}x}\right)^2}\,\mathrm{d}x.$$

Si appprossima il valore di $L$ utilizzando la formula dei trapezi composita.

1. Determinare il numero di sottointervalli sufficiente a garantire che il valore approssimato $\tilde{L}$ soddisfi
$$|L - \tilde{L}| \leq 2^{-34}.$$
2. Scrivere una funzione Matlab® che dati in input $N \in \mathbb{N}$ restituisce in output l'approssimazione $\tilde{L}(N)$ di $L$ ottenuta mediante la formula dei trapezi composita su $N$ sottointervalli.
3. Si calcoli il valore di $L$ utilizzando il comando `integral`.
4. Mediante il comando `plot` tracciare il grafico di $s(N) = |(\tilde{L}(2N) - L)/(\tilde{L}(N) - L)|$, $1 \leq N \leq 100$.
5. Commentare il risultato indicando se e perché in accordo con le stime teoriche.

**Esercizio 73.** Si denota con $I(N)$ il valore restituito dalla formula dei trapezi iterata su $N$ sottointervalli per l'approssimazione dell' integrale $I = \int_0^1 e^{-x^2}\,\mathrm{d}x$.

1. Determinare un numero $N \in \mathbb{N}$ sufficiente a garantire che $|I(N) - I| \leq 1.0e-6$.
2. Scrivere una funzione Matlab che dato in input $N$ restituisce in output $I(N)$. Riportare $\epsilon_{128} = |I - I(128)|$ e $\epsilon_{256} = |I - I(256)|$, assumendo per l'integrale il valore $I = 7.468241807264250e-01$ restituito dal comando `quad` di MatLab..
3. Determinare i coefficienti $a_0$ e $a_1$ del polinomio $p(x) = a_0 + a_1 x^2$ tale che $p(1/128) = I(128)$ e $p(1/256) = I(256)$. Determinare $\epsilon = |a_0 - I|$ con $I$ come al punto precedente.

**Esercizio 74.** Sia

$$\mathcal{I}(x) = \int_1^5 \frac{1}{x+y}\,\frac{1}{y}\,\mathrm{d}y, \quad 1 \leq x \leq 5.$$

Siano inoltre $x_j$, $1 \leq j \leq N+1$, $N+1$ punti equispaziati nell'intervallo $[1, 5]$ con $x_1 = 1$ e $x_{N+1} = 5$ e si denoti con $\mathcal{I}_j^{(M)}$ l'approssimazione di $\mathcal{I}(x_j)$ prodotta dalla formula dei trapezi iterata su $M$ sottointervalli.

1. Si determini un valore di $M$ sufficiente a garantire
$$|\mathcal{I}(x_1) - \mathcal{I}_1^{(M)}| \leq 1.0e-4.$$

2. Si determini un valore di $M$ sufficiente a garantire

$$|\mathcal{I}(x_j) - \mathcal{I}_j^{(M)}| \leq 1.0e-4, \quad 1 \leq j \leq N+1.$$

3. Scrivere una funzione MatLab che dato in input $x$ e $M$ restituisce in output l'approssimazione $\mathcal{I}^{(M)}$ di $\mathcal{I}(x)$ generata dalla formula dei trapezi iterata su $M$ sottointervalli.

4. Noto che

$$\mathcal{I}(x) = g(x) = \frac{1}{x}\log\left(\frac{1+x}{1+\frac{x}{5}}\right), \quad 1 \leq x \leq 5,$$

si riporti il valore

$$\epsilon^{(M)} = \max_{1\leq j\leq N+1} |I_j^{(M)} - g(x_j)|,$$

per $N = 32$ e $M = 32, 64, 128$. Si calcoli inoltre

$$r_1 = \epsilon^{(32)}/\epsilon^{(64)}, \quad r_2 = \epsilon^{(64)}/\epsilon^{(128)}$$

giustificando i risultati ottenuti.

**Esercizio 75.** Assegnati i punti $x_i \in \mathbb{R}$ , $1 \leq i \leq n$, , $x_i \neq x_j$ se $i \neq j$, si definisce la sequenza di polinomi

$$p_1(x) = 1, \quad p_i(x) = \prod_{j=1}^{i-1}(x - x_j), \quad 2 \leq i \leq n+1.$$

1. Si dimostri che vale $p_{i+1}(x) = (x - x_i)p_i(x)$, $1 \leq i \leq n$.

2. Scrivere una funzione MatLab che dati in input $\boldsymbol{x} = [x_1, \ldots, x_n]$ e $z \in \mathbb{R}$ restituisce in output il vettore $\boldsymbol{p} = [p_2(z), \ldots, p_{n+1}(z)]$ delle valutazioni dei polinomi $p_i(x)$ nel punto $z$. Valutare il costo computazionale dellalgoritmo.

3. Sia $A = (a_{i,j}) \in \mathbb{R}^{nn}$ definita da $a_{i,j} = p_j(x_i)$, $1 \leq i, j \leq n$. Dimostrare che $A$ risulta invertibile.

4. Dimostrare che il seguente problema di interpolazione ammette una ed una sola soluzione: determinare $a_1, \ldots, a_n$ in modo che il polinomio

$$p(x) = \sum_{j=1}^{n} a_j p_j(x)$$

soddisfi le condizioni di interpolazione $p(x_i) = f_i$, $1 \leq i \leq n$.

**Esercizio 76.** Sia

$$f(x) = \frac{1}{1+25x^2}, \quad -1 \leq x \leq 1.$$

1. Scrivere una funzione MatLab che dato in input il naturale $n$ costruisce il vettore $x \in \mathbb{R}^n$ di $n$ punti equispaziati nell'intervallo $[-1, 1]$ e restituisce in output i coefficenti del polinomio di interpolazione sui nodi $x_i$ alla funzione $f(x)$.

2. Confrontare il grafico della funzione $f(x)$ e del polinomio di interpolazione ottenuto per $n = 20$ nell'intervallo $[-1, 1]$.

3. Scrivere una funzione Matlab® che dato in input il naturale $n$ costruisce il vettore $x \in \mathbb{R}^n$ definito da $x_i = \cos(\pi(i-1)/19)$, $1 \leq i \leq 20$, e restituisce in output i coefficenti del polinomio di interpolazione sui nodi $x_i$ alla funzione $f(x)$.

4. Confrontare il grafico della funzione $f(x)$ e del polinomio di interpolazione ottenuto per $n = 20$ nell'intervallo $[-1, 1]$.

5. Siano $a, b \in \mathbb{C}$, $a \neq b$ e definiamo

$$g(t) = \det \begin{bmatrix} 1 & 1 & 1 \\ a & b & t \\ a^2 & b^2 & t^2 \end{bmatrix}.$$

   Mostrare che $g(t)$ è un polinomio di secondo grado. Mostrare che $g(a) = g(b) = 0$. Dedurre che $g(t) = k(t-a)(t-b)$ per un'opportuna costante $k$ di cui si chiede l'espressione.

# Capitolo 12

# Politiche di Vaccinazione e Modelli Epidemiologici

Le politiche attive di vaccinazione collettiva sono finalizzate al raggiungimento di una soglia di vaccinati nella popolazione sufficiente a garantire il controllo e/o l'eradicazione della malattia. In assenza di obblighi fissati per legge la valutazione sull'opportunità di vaccinarsi dipende da una comparazione rischi/benefici. Sia indicata con $p$ ed $1-p$ rispettivamente la percentuale della popolazione di vaccinati e non vaccinati. Il rischio percepito dai vaccinati è essenzialmente indipendente dal contesto ambientale e coincide con il rischio connesso alla vaccinazione. Differentemente il rischio percepito dai non vaccinati dipende dalla probabilità di infettarsi che è ovviamente funzione di $p$. In prima approssimazione possiamo pensare che esso valga $r(p) = \alpha(1-p)$. In tali condizioni si osserva che il punto di intersezione tra la retta orizzontale che rappresenta il rischio percepito dai vaccinati e la retta obliqua che specifica il rischio percepito dai non vaccinati rappresenta un punto di equilibrio del sistema.

Per riuscire a produrre stime accurate e significative di questo punto e per valutare la soglia critica si rende necessario disporre di modelli più attendibili per la diffusione della malattia che tengano conto delle sue specificità. L'idea è quella di suddividere la popolazione at tempo $t$ in tre gruppi:

1. il gruppo dei *suscettibili* $S = S(t)$ formato da individui che non hanno contratto la malattia e sono esposti al rischio del contagio;

2. il gruppo degli *infetti* $I = I(t)$ formato da individui che hanno contratto il virus e possono trasmetterlo ad altri;

3. ed infine il gruppo dei *rimossi* $R = R(t)$ formato da individui immuni al contagio.

Il modello epidemiologico corrispondente assume l'acronimo di $SIR$ dalle iniziali dei tre gruppi. Ponendo $\beta$ il tasso di trasmissione della malattia e $\nu$ il tasso di

guarigione il modello assume la forma di un sistema di tre equazioni differenziali

$$\begin{cases} \dfrac{\mathrm{d}I}{\mathrm{d}t} = \beta SI - \nu I \\ \dfrac{\mathrm{d}R}{\mathrm{d}t} = \nu I \\ \dfrac{\mathrm{d}S}{\mathrm{d}t} = -\beta SI \end{cases}$$

dove il termine quadratico $\beta SI$ di nuovi individui infetti per unità di tempo tiene conto dell'interazione tra il gruppo dei suscettibili e di infetti e la terza equazione deriva dall'equazione $S(t) + I(t) + R(t) = N$ che esprime l'invarianza nel tempo del numero $N$ di individui della popolazione. Se supponiamo che ogni individuo infetto abbia $\kappa$ contatti nell'unità di tempo di cui $\kappa S/N$ con individui suscettibili allora detta $\tau$ la percentuale di questi contatti che risulta in un'infezione abbiamo che

$$\kappa\tau IS/N = \beta SI$$

da cui $\beta = \kappa\tau/N = b/N$. Il parametro $\tau$ è detto indice di trasmissibilità della malattia e denota la probabilità di infezione per contatto.

Dall'invarianza della popolazione segue che possiamo ridurre il modello al sistema fornito dalle equazioni (1) e (3)

$$\begin{cases} \dfrac{\mathrm{d}I}{\mathrm{d}t} = \beta SI - \nu I \\ \dfrac{\mathrm{d}S}{\mathrm{d}t} = -\beta SI. \end{cases}$$

Per il calcolo delle funzioni $I(t)$ e $S(t)$ possiamo utilizzare uno schema numerico basato sulle formule di quadratura. Assumiamo noti i valori $I_0 = I(t_0)$ e $S_0 = S(t_0)$ assunti dalle funzioni al tempo $t = t_0$. Dal teorema fondamentale del calcolo integrale segue che

$$I(t_1) - I(t_0) = \int_{t_0}^{t_1} \frac{\mathrm{d}I}{\mathrm{d}t} dt.$$

L'integrale può essere approssimato con la formula dei trapezi ottenendo

$$I_1 - I_0 = \frac{(t_1 - t_0)(\beta S_1 I_1 - \nu I_1 + \beta S_0 I_0 - \nu I_0)}{2}.$$

Analogamente abbiamo

$$S_1 - S_0 = -\frac{(t_1 - t_0)(\beta S_1 I_1 + \beta S_0 I_0)}{2}.$$

Le quantità $I_1$ e $S_1$ rappresentano le approssimazioni numeriche dei valori incogniti $I(t_1)$ e $S(t_1)$. Il sistema di due equazioni non lineari in due incognite $I_1$ e $S_1$

$$\begin{cases} 2(I_1 - I_0) - (t_1 - t_0)(\beta S_1 I_1 - \nu I_1 + \beta S_0 I_0 - \nu I_0) = 0 \\ 2(S_1 - S_0) + (t_1 - t_0)(\beta S_1 I_1 + \beta S_0 I_0) = 0 \end{cases}$$

può essere riscritto in forma equivalente

$$\begin{cases} 2(I_1 - I_0) + 2(S_1 - S_0) + h\nu(I_1 + I_0) = 0 \\ 2(S_1 - S_0) + h(\beta S_1 I_1 + \beta S_0 I_0) = 0 \end{cases}$$

da cui

$$\begin{cases} I_1 = \frac{2-h\nu}{2+h\nu} I_0 - \frac{2}{2+h\nu}\Delta S \\ 2\Delta S + h\beta\Delta S I_1 + \beta S_0 I_1 + \beta S_0 I_0 = 0 \end{cases}$$

con $h = t_1 - t_0$ e $\Delta S = S_1 - S_0$. Sostituendo $I_1$ nella seconda equazione si ottiene un'equazione di 2 grado in $\Delta S$ con due radici reali di segno discorde (osservare che i coefficienti del termine quadratico e del termine costante sono discordi). Indicando con $\tilde{S}$ la radice non positiva si può porre $S_1 = S_0 + \tilde{S}$ e quindi ricavare $I_1$. Il processo continua generando le successioni $\{S_i\}$ e $\{I_i\}$ che forniscono le approssimazioni numeriche dei valori assunti dalla funzione $S(t)$ e $I(t)$ nei punti $t_i$, $0 \leq i \leq m$.

L'analisi teorica e l'evidenza sperimentale mostrano che il comportamento della funzione $I(t)$ dipende dal parametro $R = (S(0)/N)b/\nu$. In particolare se $R(0) = 0$ e $R \leq 1$ allora $I(t)$ è monotona decrescente e $\lim_{t\to+\infty} I(t) = 0$. Se $S(0)/N \simeq 1$ e $p$ indica la percentuale della popolazione vaccinata allora $R = (1-p)b/\nu$ da cui segue che l'eradicazione dell'infezione può essere assicurata anche senza la completa copertura della popolazione. Questa osservazione evidenziata dal modello matematico è alla base delle politiche attive di salute pubblica. Se se $R(0) = 0$ e $R > 1$ allora il sistema può evolvere verso uno stato endemico stabile per cui $\lim_{t\to+\infty} S(t) = \ell > 0$. In tal caso la diffusione dell'epidemia può cessare anche senza la completa eradicazione della malattia.