

Università degli studi Milano Bicocca - Dipartimento di Fisica

# Esperimentazioni di Fisica Computazionale

S. Franceschina

June 10, 2025

## Abstract

## Contents

<b>1</b>	<b>Analisi dell'errore</b>	<b>3</b>
1.1	Teoria . . . . .	3
1.2	Esercizio 1.0.1 . . . . .	3
1.2.1	Soluzione . . . . .	3
1.3	Esercizio 1.2.1 . . . . .	3
1.3.1	Soluzione . . . . .	4
1.4	Esercizio 1.4.1 . . . . .	6
1.4.1	Soluzione . . . . .	7
1.5	Esercizio 1.4.2 . . . . .	8
1.5.1	Soluzione . . . . .	9
<b>2</b>	<b>Sistemi lineari</b>	<b>12</b>
2.1	Teoria . . . . .	12
2.2	Esercizi 2.1.1, 2.1.2, 2.1.3, 2.1.4 . . . . .	12
2.2.1	Soluzione . . . . .	13
2.3	Esercizi 2.3.1, 2.3.2, 2.3.4 . . . . .	14
2.3.1	Soluzione . . . . .	14
2.4	Esercizio 2.3.3 . . . . .	15
2.4.1	Soluzione . . . . .	15
2.5	Esercizi 2.4.1, 2.4.2 . . . . .	16
2.5.1	Soluzione . . . . .	16
2.6	Esercizio 2.5.1 . . . . .	17
2.6.1	Soluzione . . . . .	17
2.6.2	Conclusioni . . . . .	18
2.7	Esercizio 2.6.1 . . . . .	19
2.7.1	Soluzione . . . . .	19
2.8	Esercizio 2.6.2 . . . . .	19

2.8.1	Soluzione . . . . .	19
2.9	Esercizio 2.6.3 . . . . .	21
2.9.1	Soluzione . . . . .	21
2.9.2	Conclusioni . . . . .	22
<b>3</b>	<b>Radici di equazioni non lineari</b>	<b>23</b>
3.1	Teoria . . . . .	23
3.2	Esercizi 3.2.1, 3.2.2 . . . . .	23
3.2.1	Soluzione . . . . .	23
3.3	Esercizi 3.3.1, 3.3.2 . . . . .	25
3.3.1	Soluzione . . . . .	26
3.4	Esercizio 3.3.3 . . . . .	31
3.4.1	Soluzione . . . . .	31
3.5	Esercizio 3.3.4 . . . . .	32
3.6	Soluzione . . . . .	32
3.7	Esercizi 3.4.1, 3.4.2 . . . . .	33
3.7.1	Soluzione . . . . .	34
3.8	Esercizio 3.4.3 . . . . .	35
3.8.1	Soluzione . . . . .	35
<b>4</b>	<b>Interpolazioni</b>	<b>36</b>
4.1	Teoria . . . . .	36
4.2	Esercizi 4.2.1, 4.2.2 . . . . .	36
4.2.1	Soluzione . . . . .	36
<b>5</b>	<b>Integrazione numerica</b>	<b>37</b>
5.1	Teoria . . . . .	37
<b>6</b>	<b>Equazioni differenziali ordinarie</b>	<b>37</b>
6.1	Teoria . . . . .	37
<b>7</b>	<b>Appendice</b>	<b>37</b>
7.1	Dati . . . . .	37

# 1 Analisi dell'errore

## 1.1 Teoria

Nella presente sezione analizziamo le due principali fonti di errore in contesti computazionali:

1. Errori di arrotondamento: dovuti alla rappresentazione di numeri reali con numero finito di digits.
2. Errori di approssimazione: dovuti alla modalità stessa con cui affrontiamo il problema, per questo motivo sono presenti anche nel caso ideale.

## 1.2 Esercizio 1.0.1

Considera la funzione  $f(x) = e^x$  nell'intervallo  $x \in [0, 1]$ . Scrivi un programma che calcoli la serie approssimante:

$$g_N(x) = \sum_{n=0}^N \frac{x^n}{n!}. \quad (1)$$

1. Verifica che l'errore assoluto  $\Delta = |f(x) - g_N(x)|$  scala approssimativamente come  $x^{N+1}/(N+1)!$  per  $N = 1, 2, 3, 4$ .
2. L'errore  $\Delta$ , nell'intervallo dato di  $x$ , differisce da  $x^{N+1}/(N+1)!$ . Perché accade questo e per quali valori di  $x$ ?

### 1.2.1 Soluzione

Al fine dell'esercizio vengono rappresentati nel grafico 1 le funzioni  $\Delta$  e  $\frac{x^{N+1}}{(N+1)!}$ , con  $N = 1, 2, 3, 4$ , al variare di  $x$  nell'intervallo  $[0, 1]$ .

La prima considerazione che possiamo fare è che all'aumentare di  $N$  la funzione  $\Delta$  assume valori sempre più vicini allo zero. Questo significa che la distanza tra il valore della funzione  $f(x)$  presa in esame e la sua espansione di Taylor troncata all'ordine  $N$  diminuisce. In effetti ci aspettiamo che la funzione  $\Delta$  sia esattamente zero nel caso in cui  $N \rightarrow \infty$ . Inoltre possiamo notare che la funzione  $\Delta$ , in ognuno dei grafici, è tanto più prossima allo zero quanto più ci si avvicina all'origine, poichè l'espansione in serie richiede  $x \rightarrow 0$ .

La seconda considerazione è che le funzioni  $\Delta$  e  $\frac{x^{N+1}}{(N+1)!}$  si avvicinano tra loro all'aumentare di  $N$ . Questo risponde alla richiesta dell'esercizio, cioè che l'errore scali come un polinomio di ordine  $N + 1$ .

## 1.3 Esercizio 1.2.1

Calcolare la seguente somma

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6} = \lim_{N \rightarrow \infty} S(N) \quad \text{con} \quad S(N) = \sum_{n=1}^N \frac{1}{n^2} \quad (2)$$

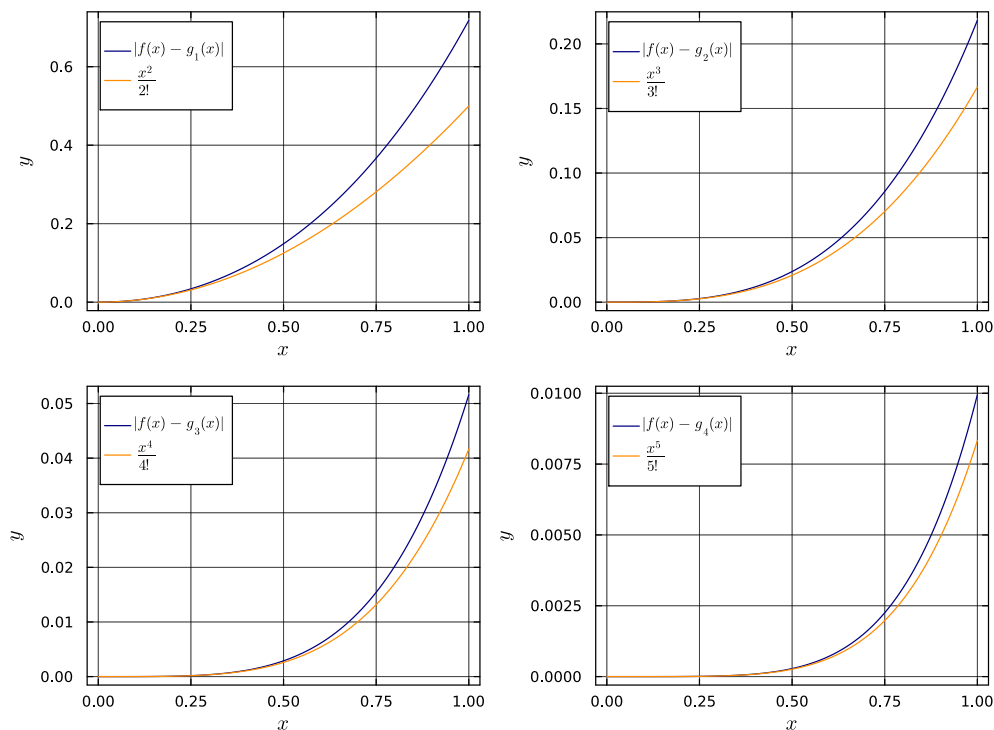


Figure 1: Confronto tra  $\Delta$  e  $\frac{x^{N+1}}{(N+1)!}$  per  $N = 1, 2, 3, 4$ .

1. Calcolare la somma in single precision utilizzando l'ordinamento normale,  $n = 1, 2, 3, \dots, N$ .
2. Calcolare la somma in single precision utilizzando l'ordinamento inverso,  $n = N, \dots, 2, 1$ .
3. Studiare la convergenza di entrambe le implementazioni in funzione di  $N$  tracciando il grafico di  $|S(N) - \pi^2/6|$ .
4. Ripetere i punti da 1 a 3 utilizzando double precision.
5. Sai spiegare cosa succede?

### 1.3.1 Soluzione

Procediamo prendendo in esame i due casi, single precision e double precision.

**Single precision:** Si riportano in figura 2 gli errori di troncamento della serie  $\sum_{n=1}^{\infty} \frac{1}{n^2}$ , laddove  $N$  è il troncamento. Le somme sono state eseguite con variabili di tipo Float32 (single precision).

In figura 2 notiamo due comportamenti degni di nota. Il primo riguarda la curva blu, che arresta la sua discesa poco dopo  $N = 4000$ . Il secondo riguarda il discostarsi delle due curve, già a partire da  $N \simeq 2000$ , a causa dell'andamento a linea spezzata della curva blu.

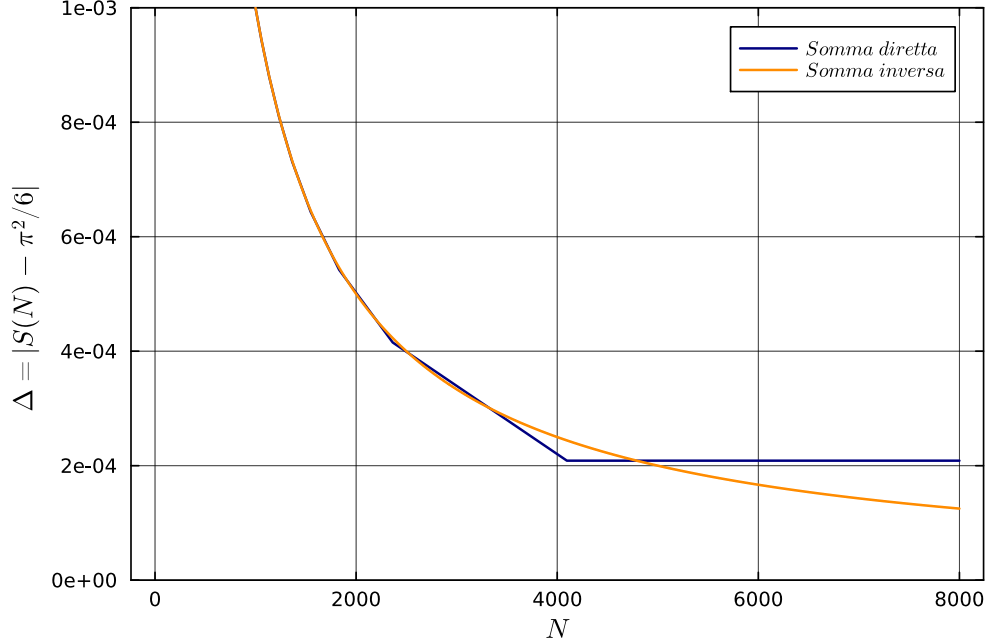


Figure 2:  $S(N) = \sum_{n=1}^N \frac{1}{n^2}$  in single precision

La spiegazione del primo fenomeno è la seguente: nel caso della somma con ordinamento diretto, all'aumento dell'indice di somma, gli addendi sono sempre più piccoli. In particolare la somma comincia da 1, e sappiamo che dovrà raggiungere  $\frac{\pi^2}{6} \simeq 1.64$  e perciò per tutto il processo la somma parziale resterà nell'intervallo  $[1, 2)$ . In tale intervallo la distanza tra due floating point numbers è  $\epsilon_{mach} = 2^{-23}$ . Osserviamo il grafico 2: la curva blu comincia ad essere costante a partire da  $N = 4096 = 2^{12}$ . Tale numero corrisponde all'addendo  $\frac{1}{N^2} = 2^{-24}$ , che è appena più piccolo di  $\epsilon_{mach}$ , ovvero della distanza tra due floating point numbers in  $[1, 2)$ , e quindi è come sommare 0.

La spiegazione del secondo fenomeno è simile: in questo caso alla somma viene aggiunto un addendo che è più piccolo del precedente, ma non così tanto da essere più piccolo di  $\epsilon_{mach}$ . In questo modo la somma viene migliorata, ma il nuovo addendo è arrotondato rispetto al suo valore vero. Il successivo addendo, pur essendo differente in teoria, a causa dell'arrotondamento risulta essere uguale al precedente. Ciò fa in modo che venga sommato sempre lo stesso numero, e così l'andamento della curva blu è lineare.

I due fenomeni non si verificano per la somma con ordinamento inverso perchè il primo numero ad essere sommato è molto piccolo. Questo fa in modo che i successivi floating point number siano molto vicini tra loro, è così sommare l'addendo successivo fa cadere la somma parziale vicina al floating point number che la approssima.

Si può mostrare matematicamente che gli errori  $\Delta = |S(N) - \pi^2/6|$  scalano come  $O(\frac{1}{N})$ . Per verificare questa affermazione possiamo interpolare la curva degli errori con un metodo che vedremo nella sezione 2. Sulla base delle considerazioni di questa sezione, possiamo aspettarci che sia più conveniente interpolare la curva con ordinamento inverso, perchè i valori che la compongono sono meno affetti da errori di tipo numerico. Utilizzando il modello lin-

earizzato  $\log(\Delta) = -c * \log(N)$  con  $c = 1$  si ottiene per l'ordinamento inverso  $c_{fit} = 1.00003$  e per quello diretto  $c_{fit} = 0.986$ . Come atteso il caso di ordinamento diretto restituisce un risultato peggiore di quello inverso.

**Double precision:** Si riporta in figura 3 un ingrandimento del grafico degli errori di troncamento della serie  $\sum_{n=1}^{\infty} \frac{1}{n^2}$ , laddove  $N$  è il troncamento. La somme sono state eseguite con variabili di tipo Float64 (double precision).

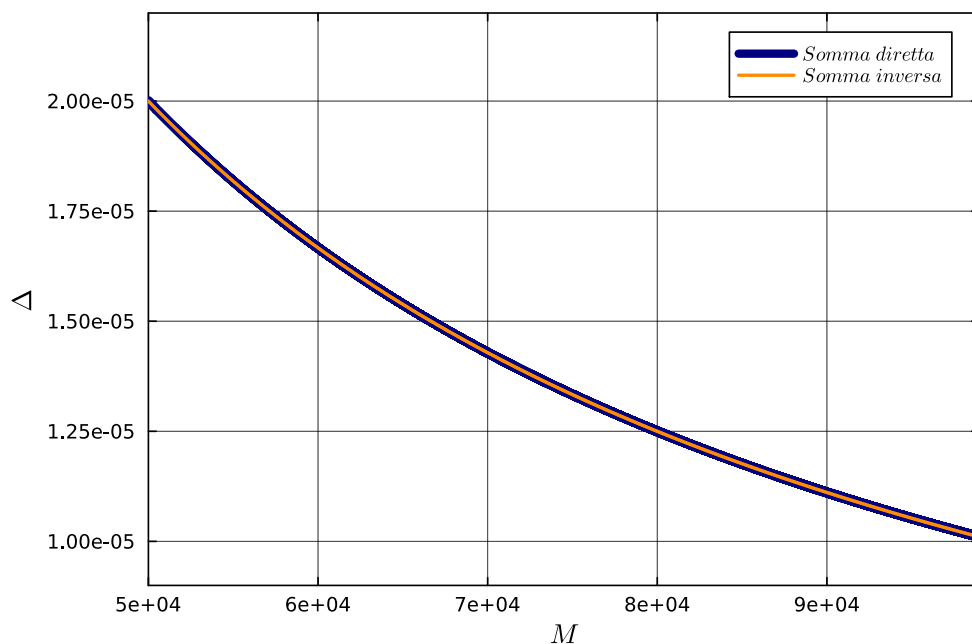


Figure 3: Ingrandimento di  $S(N) = \sum_{n=1}^N \frac{1}{n^2}$  in double precision

Nel caso di figura 3 possiamo notare che le due curve sono sovrapposte, nonostante l'ingrandimento a valori di  $N \in [5000, 10000]$ . Per osservare un fenomeno simile a quello di figura 2 dovremmo raggiungere valori di  $N = 2^{26}$ . Infatti l'approssimazione non migliora quando vengono sommati numeri dell'ordine di  $\epsilon_{mach}$  per double precision, cioè  $\frac{1}{N^2} \simeq 2^{-52}$ .

## 1.4 Esercizio 1.4.1

(a) In statistica, definiamo la varianza di un campione di valori  $x_1, \dots, x_n$  come

$$\sigma^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2, \quad \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i. \quad (3)$$

Scrivi una funzione che prenda in input un vettore  $x$  di lunghezza arbitraria e restituisca  $\sigma^2$  calcolata con la formula sopra. Dovresti testare la funzione con  $x = [1, 1, \dots, 1]$  e con alcuni vettori casuali.

(b) Uno svantaggio della formula "naive" è che bisogna prima calcolare una somma per  $\bar{x}$  prima di eseguire la somma per calcolare  $\sigma^2$ . Questo significa che bisogna scorrere due volte i dati. Ciò è indesiderabile per grandi insiemi di dati o quando la varianza campionaria deve essere calcolata mentre i dati vengono generati. Alcuni libri di testo riportano una formula a singolo ciclo:

$$\sigma^2 = \frac{1}{n-1} \left( u - \frac{1}{n} v^2 \right), \quad u = \sum_{i=1}^n x_i^2, \quad v = \sum_{i=1}^n x_i. \quad (4)$$

Prova entrambe le formule per i seguenti dataset, ciascuno dei quali ha varianza esattamente uguale a 1. Esegui i calcoli sia in single che in double precision.

$x_1 = [1e3, 1 + 1e3, 2 + 1e3]$

$x_2 = [1e6, 1 + 1e6, 2 + 1e6]$

$x_3 = [1e7, 1 + 1e7, 2 + 1e7]$

$x_4 = [1e8, 1 + 1e8, 2 + 1e8]$

Sai spiegare i risultati?

#### 1.4.1 Soluzione

(a) Si è testato il codice che implementa la formula 3 su tre vettori contenenti elementi di tipo Float64. Il primo contiene solo numeri 1, il secondo contiene numeri generati a partire da una distribuzione uniforme, il terzo numeri generati a partire da una distribuzione normale. I valori sono riportati in tabella 1.

Table 1: Risultati del test della formula 3 su tre vettori di tipo Float64

Vettore	Lunghezza	Varianza attesa	Varianza (output)
Solo uno	$10^3$	0.000	0.000
Uniforme	$10^8$	0.083	0.083
Normale	$10^8$	1.000	1.000

I risultati di tabella 1 mostrano che l'implementazione del codice è corretta.

(b) Si è testato il codice che implementa la formula 4 su quattro vettori contenenti elementi di tipo Float32, Float64 e Long Double. I risultati sono riportati in tabella 2.

Table 2: Valori di varianza. Metodo 1: equazione 3. Metodo 2: equazione 4.

Set	$\sigma^2$ attesa	Float32		Float64		Long Double	
		Metodo 1	Metodo 2	Metodo 1	Metodo 2	Metodo 1	Metodo 2
$x_1$	1.0	1.0	1.0	1.0	1.0	1.0	1.0
$x_2$	1.0	1.0	-131072.0	1.0	1.0	1.0	1.0
$x_3$	1.0	1.0	0.0	1.0	1.0	1.0	1.0
$x_4$	1.0	0.0	0.0	1.0	0.0	1.0	1.0

I risultati di tabella 2 mostrano che il calcolo della varianza con la formula 4 è più affetto dalla mancanza di precisione della rappresentazione dei numeri durante il calcolo. Infatti,

mano a mano che la la precisione dei numeri aumenta, il risultato della formula 4 si avvicina a quello atteso.

C'è da notare che anche la formula 3 è affetta dalla stessa problematica, ma in maniera meno evidente. Basti osservare il risultato del metodo 1 per il vettore  $x_4$  in Float32, che è 0.0, contro l'atteso 1.0.

Parte del problema risiede nella cancellazione numerica. Infatti, per valori degli elementi del vettore molto grandi, la differenza tra il quadrato della somma e la somma dei quadrati è molto piccola e incorre in problema di cancellazione. Per verificare questa affermazione, si riportano in tabella 3 i valori della differenza  $u - \frac{v^2}{n}$  per la formula 4.

Table 3: Differenza  $u - \frac{v^2}{n}$  per ciascun dataset.

Dataset	Single	Double	Long Double
$x_1$	2.0	2.0	2.0
$x_2$	-262144.0	2.0	2.0
$x_3$	0.0	2.0	2.0
$x_4$	0.0	0.0	2.0

Leggendo la tabella 3 possiamo notare che per il dataset  $x_1$  la differenza è esattamente 2.0, che è il risultato atteso. Per il dataset  $x_2$  la differenza è molto grande,  $-262144.0$ , il che indica un significativo problema di cancellazione numerica. Per il dataset  $x_3$  e  $x_4$ , la differenza è 0.0, indice del fatto che i singoli valori  $u - \frac{v^2}{n}$  sono vicini tra loro meno di  $\epsilon_{mach}$ , e perciò la loro differenza è zero.

Per concludere, possiamo dire che la formula 4 è più efficiente in termini di tempo di calcolo, ma è più affetta da errori numerici rispetto alla formula 3.

## 1.5 Esercizio 1.4.2

Sia  $f(x) = \frac{e^x - 1}{x}$ .

(a) Trova il numero di condizionamento  $\kappa_f(x)$ . Qual è il massimo di  $\kappa_f(x)$  nell'intervallo  $-1 \leq x \leq 1$ ?

(b) Usa l'algoritmo "naive"

$$f(x) = \frac{e^x - 1}{x} \quad (5)$$

per calcolare  $f(x)$  per  $x = 10^{-3}, 10^{-4}, 10^{-5}, \dots, 10^{-16}$ .

(c) Crea un secondo algoritmo utilizzando i primi  $n$  termini della serie di Maclaurin, cioè

$$p(x) = 1 + \frac{1}{2!}x + \frac{1}{3!}x^2 + \dots + \frac{1}{(n+1)!}x^n. \quad (6)$$

Valutalo sugli stessi valori di  $x$  del punto (b). Per farlo devi scegliere un valore per  $n$ . Verifica la stabilità del risultato al variare di  $n$ . Avresti potuto indovinare un buon valore di  $n$  fin dall'inizio?

(d) Confronta i risultati delle due implementazioni in funzione di  $x$ . Quale algoritmo pensi sia più accurato, e perché?



### 1.5.1 Soluzione

(a) Il numero di condizionamento di una funzione  $f$  è definito come:

$$\kappa_f(x) = \left| \frac{xf'(x)}{f(x)} \right|. \quad (7)$$

Per la funzione  $f(x) = \frac{e^x - 1}{x}$ , calcoliamo la derivata:

$$f'(x) = \frac{e^x x - (e^x - 1)}{x^2} = \frac{e^x(x - 1) + 1}{x^2}. \quad (8)$$

Quindi il numero di condizionamento diventa:

$$\kappa_f(x) = \left| \frac{x \left( \frac{e^x(x-1)+1}{x^2} \right)}{\frac{e^x-1}{x}} \right| = \left| \frac{e^x(x-1)+1}{(e^x-1)x} \right| = \left| \frac{e^x x}{(e^x-1)} - 1 \right|. \quad (9)$$

La derivata di  $k_f(x)$  non si annulla mai nell'intervallo  $-1 \leq x \leq 1$ , e  $\kappa_f(0) = 0$ . Dato che  $\kappa_f(x)$  è monotona crescente per  $x \geq 0$  e monotona decrescente per  $x < 0$ , il massimo si ha in  $x = 1$ , dove assume il valore  $\kappa_f(1) = 0.58198$ .

**(b)-(c):** Si riportano in figura 4 i risultati dei due algoritmi. La dicitura  $f(x)$  si riferisce alla funzione calcolata con l'algoritmo "naive", mentre  $p_n(x)$  si riferisce alla funzione calcolata con la serie di Maclaurin, fino al termine  $n$ .

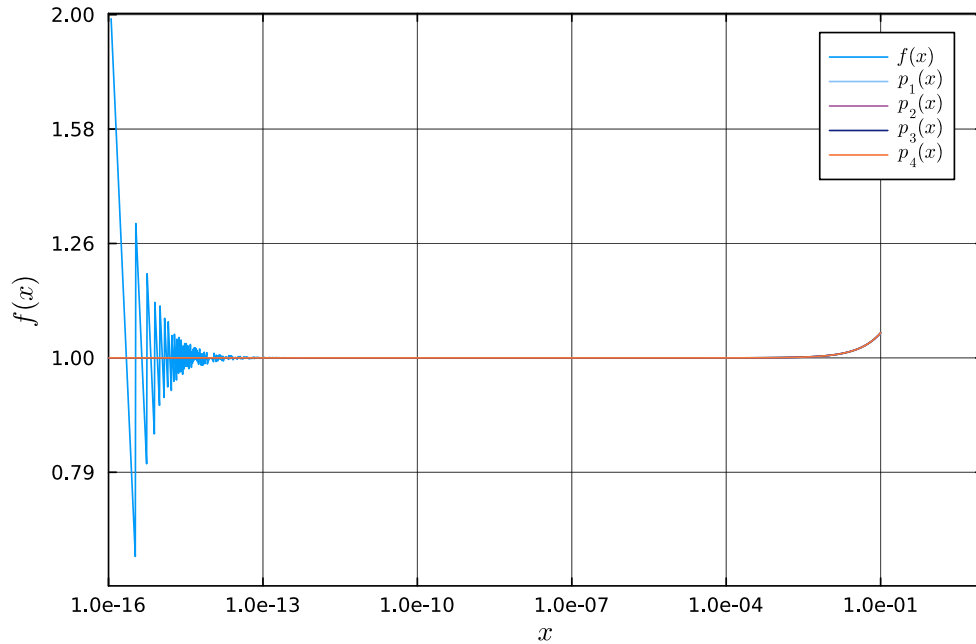


Figure 4: Confronto tra  $f(x)$  e  $p_n(x)$  al variare di  $x$ .

Per uno studio più accurato, si riporta in figura 5 un ingrandimento per piccoli valori di  $x$  (range  $[10^{-15}, 10^{-14}]$ ) della figura 4 e in figura 6 un ingrandimento per valori di  $x$  (range  $[10^{-2}, 10^{-1}]$ ) della figura 4.

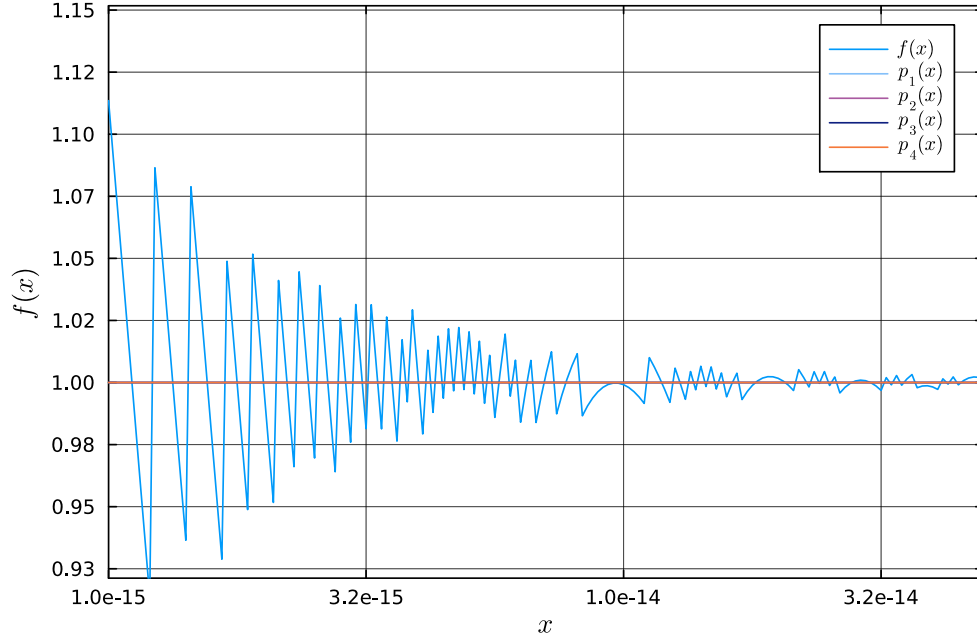


Figure 5: Ingrandimento di  $f(x)$  e  $p_n(x)$  per piccoli valori di  $x$ .

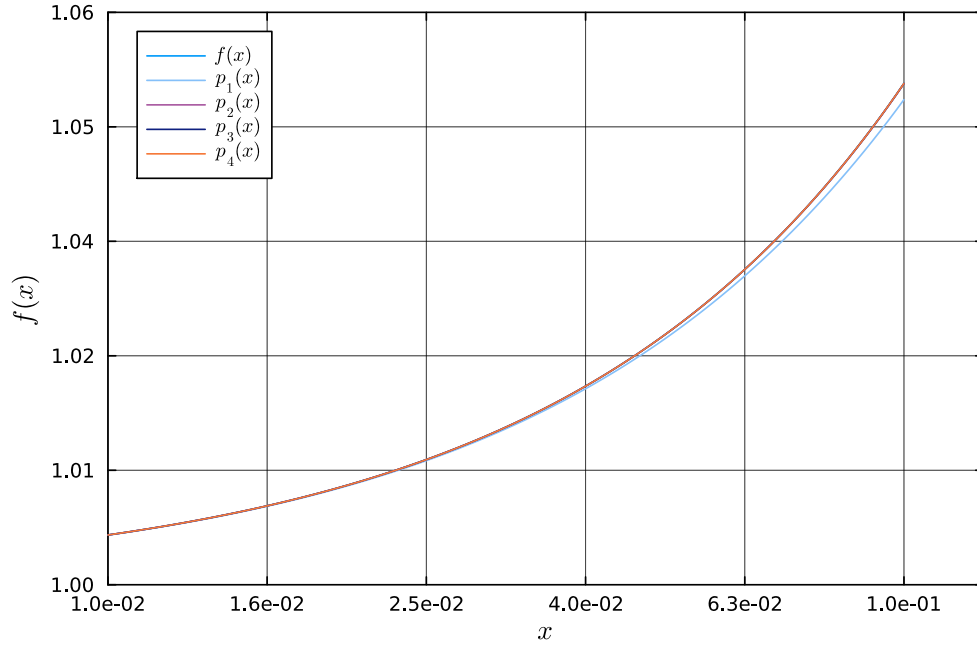


Figure 6: Ingrandimento di  $f(x)$  e  $p_n(x)$  per valori di  $x$  nell'intervallo  $[10^{-2}, 10^{-1}]$ .

Per quanto riguarda il comportamento della funzione  $f(x)$  calcolata con l'algoritmo 5, possiamo notare che per valori di  $x$  piccoli, in particolare nella regione rappresentata in figura 5, la funzione assume valori non corretti, nonostante  $\kappa_f(x)$  abbia un massimo non molto grande e peraltro non assunto nell'intorno di zero. La spiegazione di questo fenomeno

è che l'algoritmo 5, nonostante non abbia  $\kappa_f(x)$  complessivo molto grande, è affetto da errori di cancellazione numerica a causa della forma in cui è scritta la funzione, ma non a causa del fatto che il problema sia intrinsecamente instabile. Di fatto possiamo intuire che esista una possibile riscrittura della funzione che non sia affetta da cancellazione numerica, e che sia più stabile, proprio perchè sappiamo che il massimo di  $\kappa_f(x)$  non è molto grande. Tale riscrittura è quella che utilizza la serie di Maclaurin, implementata nell'algoritmo  $p_n(x)$ .

Possiamo notare che l'algoritmo 6, per valori di  $x$  piccoli, restituisce il valore corretto, dato che non è affetto da cancellazione numerica. Al contrario, nella regione ingradata di figura 6, l'algoritmo 6 comincia a restituire valori errati perchè ci allontaniamo dalla regione di convergenza della serie di Maclaurin.

Osservando la figura 6, notiamo che, per valori di  $x$  fissati, l'algoritmo 6 è stabile, cioè le curve che descrivono i valori di  $p_n(x)$  sono sempre più vicine tra loro all'aumentare di  $n$ , fino a sovrapporsi. Per stimare un buon valore di  $n$  da utilizzare per calcolare  $p_n(x)$ , possiamo osservare che la serie di Maclaurin è tanto meno stabile quanto più ci si allontana da zero. Nel caso di studio valutiamo la distanza tra i valori di  $p_n(x)$  al variare di  $n$ , fissato  $x = 0.1$ . Si è rappresentato in figura 7 l'andamento della distanza tra i valori di  $p_n(0.1)$  e  $p_{n+1}(0.1)$  al variare di  $n$ . In formula:

$$\Delta_n = |p_n(0.1) - p_{n+1}(0.1)|. \quad (10)$$

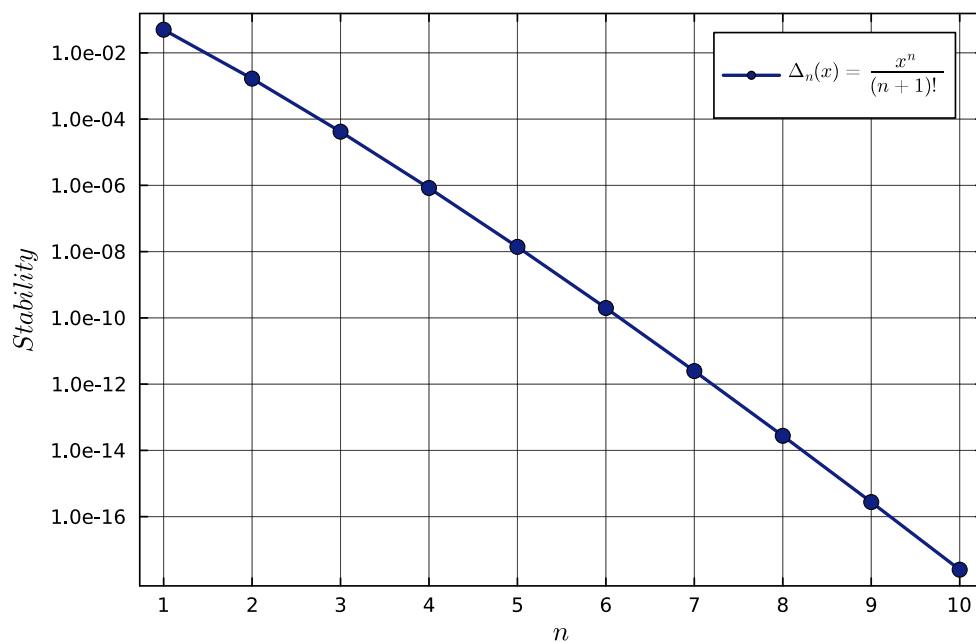


Figure 7: Distanza tra  $p_n(0.1)$  e  $p_{n+1}(0.1)$  al variare di  $n$ .

Ai fini di questo esercizio possiamo dire che  $n = 2$  è già un buon valore perchè le distanze tra i polinomi successivi non sono apprezzabili nel range  $[1, 1.06]$ , caratteristico della figura 6. È chiaro che volendo raggiungere precisione massima occorrerà scegliere  $n = 10$ , dato che  $\Delta_n$  corrispondente è più piccolo di  $\epsilon_{mach}$

(d): Come confronto tra i due algoritmi si riporta in figura 7 il grafico dell'errore tra i due algoritmi, cioè

$$\Delta(x) = |f(x) - p_n(x)| . \quad (11)$$

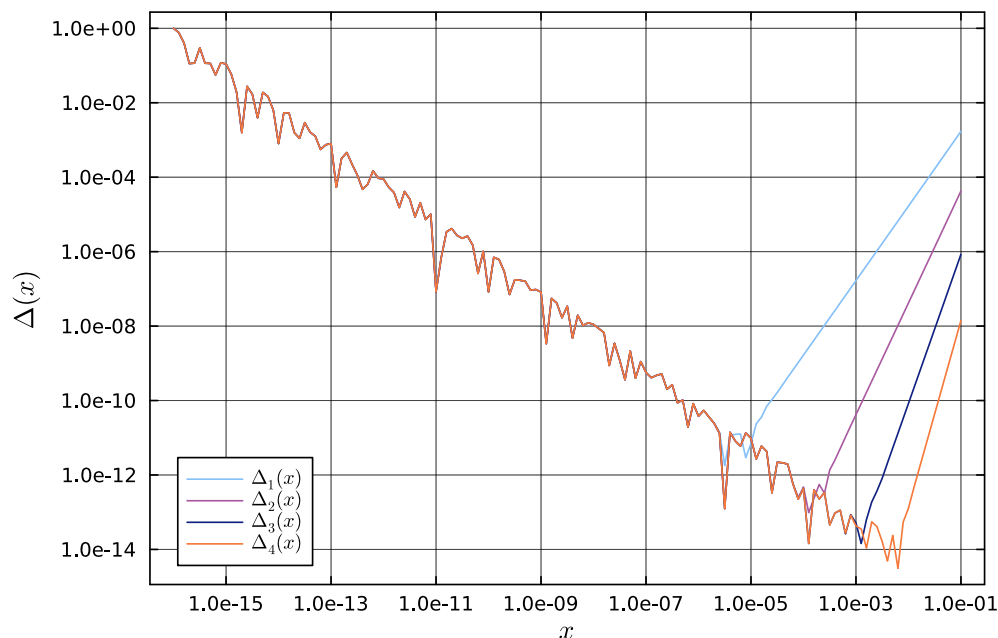


Figure 8: Distanza tra  $f(x)$  e  $p_n(x)$ .

La figura 8 mostra una evidente differenza tra i due algoritmi nella regione di instabilità numerica di dell'algoritmo 5, cioè per valori di  $x$  piccoli. E' interessante notare che la distanza tra i due algoritmi diminuisce all'aumentare di  $x$ , ma torna ad aumentare a partire da  $x = 10^{-5}$ , laddove l'algoritmo 6 esce dalla regione di convergenza della serie di Maclaurin. In conclusione, possiamo affermare che l'algoritmo 6 è più accurato per valori di  $x$  piccoli, mentre l'algoritmo 5 è più accurato per valori di  $x$  grandi.

## 2 Sistemi lineari

### 2.1 Teoria

### 2.2 Esercizi 2.1.1, 2.1.2, 2.1.3, 2.1.4

Scrivi una funzione che esegua la sostituzione in avanti su una matrice triangolare inferiore  $n \times n$ .

2. Testa il tuo codice sui seguenti sistemi lineari:

$$(a) \begin{bmatrix} -2 & 0 & 0 \\ 1 & -1 & 0 \\ 3 & 2 & 1 \end{bmatrix} \mathbf{x} = \begin{bmatrix} -4 \\ 2 \\ 1 \end{bmatrix}$$

$$(b) \begin{bmatrix} 4 & 0 & 0 & 0 \\ 1 & -2 & 0 & 0 \\ -1 & 4 & 4 & 0 \\ 2 & -5 & 5 & 1 \end{bmatrix} \mathbf{x} = \begin{bmatrix} -4 \\ 1 \\ -3 \\ 5 \end{bmatrix}$$

Puoi verificare la soluzione risolvendo il sistema a mano e/o calcolando  $\mathbf{L}\mathbf{x} - \mathbf{b}$ .

3. Scrivi una funzione che esegua la sostituzione all'indietro su una matrice triangolare superiore  $n \times n$ .

4. Testa il tuo codice sui seguenti sistemi lineari:

$$(a) \begin{bmatrix} 3 & 1 & 0 \\ 0 & -1 & -2 \\ 0 & 0 & 3 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 1 \\ 1 \\ 6 \end{bmatrix} \qquad (b) \begin{bmatrix} 3 & 1 & 0 & 6 \\ 0 & -1 & -2 & 7 \\ 0 & 0 & 3 & 4 \\ 0 & 0 & 0 & 5 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 4 \\ 1 \\ 1 \\ 5 \end{bmatrix}$$

Puoi verificare la soluzione risolvendo il sistema a mano e/o calcolando  $\mathbf{U}\mathbf{x} - \mathbf{b}$ .

### 2.2.1 Soluzione

Per risolvere i sistemi lineari proposti, si sono implementate le funzioni di sostituzione in avanti e all'indietro. Di seguito sono riportati i risultati delle soluzioni.

#### Sostituzione in avanti:

- Per il sistema (a) si ha  $\mathbf{x} = [-4.0 \ 2.0 \ 1.0]$ .
- Per il sistema (b) si ha  $\mathbf{x} = [-4.0 \ 1.0 \ -3.0 \ 5.0]$ .

Risolvendo i sistemi a mano, i valori sono:

- Per il sistema (a) si ha  $\mathbf{x} = [-4 \ 2 \ 1]$ .
- Per il sistema (b) si ha  $\mathbf{x} = [-4 \ 1 \ -3 \ 5]$ .

#### Sostituzione all'indietro:

- Per il sistema (a) si ha  $\mathbf{x} = [1.0 \ 1.0 \ 6.0]$ .
- Per il sistema (b) si ha  $\mathbf{x} = [4.0 \ 1.0 \ 1.0 \ 5.0]$ .

Risolvendo i sistemi a mano, i valori sono:

- Per il sistema (a) si ha  $\mathbf{x} = [1 \ 1 \ 6]$ .
- Per il sistema (b) si ha  $\mathbf{x} = [4 \ 1 \ 1 \ 5]$ .

I risultati confermano la correttezza delle implementazioni delle funzioni di sostituzione in avanti e all'indietro.

## 2.3 Esercizi 2.3.1, 2.3.2, 2.3.4

1. Scrivi una funzione che esegua la fattorizzazione LU di una matrice  $n \times n$ .
2. Per ciascuna matrice, calcola la fattorizzazione LU e verifica la correttezza.

(a)	(b)	(c)
$\begin{bmatrix} 2 & 3 & 4 \\ 4 & 5 & 10 \\ 4 & 8 & 2 \end{bmatrix}$	$\begin{bmatrix} 6 & -2 & -4 & 4 \\ 3 & -3 & -6 & 1 \\ -12 & 8 & 21 & -8 \\ -6 & 0 & -10 & 7 \end{bmatrix}$	$\begin{bmatrix} 1 & 4 & 5 & -5 \\ -1 & 0 & -1 & -5 \\ 1 & 3 & -1 & 2 \\ 1 & -1 & 5 & -1 \end{bmatrix}$

4. Calcola il determinante delle matrici dell'Esercizio 2 utilizzando la loro fattorizzazione LU.

### 2.3.1 Soluzione

**(Punti 1 e 2: )** Si sono implementati gli algoritmi di fattorizzazione LU, e si sono testati sui tre sistemi proposti.

Le scomposizioni ottenute sono le seguenti:

$\text{(A)} \quad \mathbf{L}_A = \begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 2.0 & 1.0 & 0.0 \\ 2.0 & -2.0 & 1.0 \end{bmatrix}$	$\mathbf{U}_A = \begin{bmatrix} 2.0 & 3.0 & 4.0 \\ 0.0 & -1.0 & 2.0 \\ 0.0 & 0.0 & -2.0 \end{bmatrix}$
$\text{(B)} \quad \mathbf{L}_B = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 \\ 0.5 & 1.0 & 0.0 & 0.0 \\ -2.0 & -2.0 & 1.0 & 0.0 \\ -1.0 & 1.0 & -2.0 & 1.0 \end{bmatrix}$	$\mathbf{U}_B = \begin{bmatrix} 6.0 & -2.0 & -4.0 & 4.0 \\ 0.0 & -2.0 & -4.0 & -1.0 \\ 0.0 & 0.0 & 5.0 & -2.0 \\ 0.0 & 0.0 & 0.0 & 8.0 \end{bmatrix}$
$\text{(C)} \quad \mathbf{L}_C = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 \\ -1.0 & 1.0 & 0.0 & 0.0 \\ 1.0 & -0.25 & 1.0 & 0.0 \\ 1.0 & -1.25 & -1.0 & 1.0 \end{bmatrix}$	$\mathbf{U}_C = \begin{bmatrix} 1.0 & 4.0 & 5.0 & -5.0 \\ 0.0 & 4.0 & 4.0 & -10.0 \\ 0.0 & 0.0 & -2.0 & 6.0 \\ 0.0 & 0.0 & 0.0 & 2.0 \end{bmatrix}$

Per verificare la correttezza delle scomposizioni, si è calcolato il prodotto  $\mathbf{LU}$  per ciascuna matrice e lo si è sottratto alla matrice originale. Si sono ottenute matrici di zeri, in tutti i casi, confermando la correttezza dell'algoritmo.

**(Punto 4: )** Si è calcolato il determinante delle matrici dell'esercizio 2 utilizzando la loro fattorizzazione LU. Ricordando le proprietà del determinante e che la matrice  $\mathbf{L}$  è triangolare inferiore con elementi diagonali unitari, si ha che:

$$\det(M) = \det(LU) = \det(\mathbf{L}) \cdot \det(\mathbf{U}) = \det(\mathbf{U}). \quad (12)$$

Per il calcolo del determinante delle matrici proposte si è eseguito il prodotto degli elementi della diagonale principale della matrice  $\mathbf{U}$ . A titolo di confronto si è calcolato il determinante delle matrici con la funzione `det` di Julia. I risultati sono riportati in tabella 4.

Table 4: Determinanti delle matrici dell'esercizio 2.

Matrice	Determinante (LU)	Determinante (Julia)
A	4.0	4.0
B	-480.0	-480.0
C	80.0	80.0

Si ottengono risultati identici.

## 2.4 Esercizio 2.3.3

Le matrici

$$\mathbf{T}(x, y) = \begin{bmatrix} 1 & 0 & x \\ 0 & 1 & y \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{R}(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

sono usate per rappresentare traslazioni e rotazioni di punti del piano in computer grafica. Per quanto segue, sia

$$\mathbf{A} = \mathbf{T}(3, -1) \mathbf{R}(\pi/5) \mathbf{T}(-3, 1), \quad \mathbf{z} = \begin{bmatrix} 2 \\ 2 \\ 1 \end{bmatrix}.$$

- (a) Calcolare  $\mathbf{b} = \mathbf{Az}$ .
- (b) Trovare la fattorizzazione LU di  $\mathbf{A}$ .
- (c) Usare i fattori e le sostituzioni triangolari per risolvere  $\mathbf{Ax} = \mathbf{b}$  e calcolare  $\mathbf{x} - \mathbf{z}$ .

### 2.4.1 Soluzione

(a) Si calcola il prodotto tra le matrici  $\mathbf{A}$  e  $\mathbf{z}$ , ottenendo:

$$\mathbf{b} = \mathbf{Az} = \begin{bmatrix} 3.95 \\ 2.01 \\ 1.0 \end{bmatrix}. \quad (13)$$

(b) Si calcola la fattorizzazione LU della matrice  $\mathbf{A}$ , ottenendo:

$$\mathbf{L} = \begin{bmatrix} 1.0 & 0.0 & 0.0 \\ -0.8 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} 0.81 & 0.59 & 1.16 \\ 0.0 & 1.24 & 2.42 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}. \quad (14)$$

(c) Si risolve il sistema  $\mathbf{Ax} = \mathbf{b}$  utilizzando le sostituzioni triangolari. Si ottiene, con le dovute approssimazioni:

$$\mathbf{x} = \begin{bmatrix} 2.0 \\ 2.0 \\ 1.0 \end{bmatrix}. \quad (15)$$

La differenza tra  $\mathbf{x}$  e  $\mathbf{z}$  è:

$$\mathbf{x} - \mathbf{z} = \begin{bmatrix} -2.22 \cdot 10^{-16} \\ -4.44 \cdot 10^{-16} \\ 0.0 \end{bmatrix}. \quad (16)$$

## 2.5 Esercizi 2.4.1, 2.4.2

1. Scrivi un programma che esegua la decomposizione di Cholesky di una matrice  $n \times n$ .
2. Per ciascuna matrice, utilizza la decomposizione di Cholesky per determinare se è definita positiva.

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 4 & 5 \\ -1 & 5 & 10 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 4 & 5 \\ 1 & 5 & 10 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 4 & 5 \\ 1 & 5 & 1 \end{bmatrix}$$

$$\mathbf{D} = \begin{bmatrix} 6 & 2 & 1 & 0 \\ 2 & 6 & 2 & 1 \\ 1 & 2 & 5 & 2 \\ 0 & 1 & 2 & 4 \end{bmatrix}$$

$$\mathbf{E} = \begin{bmatrix} 4 & 1 & 2 & 7 \\ 1 & 1 & 3 & 1 \\ 2 & 3 & 5 & 3 \\ 7 & 1 & 3 & 1 \end{bmatrix}$$

### 2.5.1 Soluzione

Si è implementato l'algoritmo di decomposizione di Cholesky, e si sono testate le matrici proposte. L'algoritmo è giunto a conclusione per le matrici  $\mathbf{A}$ ,  $\mathbf{B}$  e  $\mathbf{D}$ , ma ha restituito un messaggio di errore per le matrici  $\mathbf{C}$  e  $\mathbf{E}$ , indicando che non sono definite positive. Si riportano i risultati delle decomposizioni giunte a termine, con i dovuti arrotondamenti.

$$\mathbf{R}_\mathbf{A} = \begin{bmatrix} 1.0 & 0.0 & -1.0 \\ 0.0 & 2.0 & 2.5 \\ 0.0 & 0.0 & 1.7 \end{bmatrix}$$

$$\mathbf{R}_\mathbf{B} = \begin{bmatrix} 1.0 & 0.0 & 1.0 \\ 0.0 & 2.0 & 2.5 \\ 0.0 & 0.0 & 1.7 \end{bmatrix}$$

$$\mathbf{R}_\mathbf{D} = \begin{bmatrix} 2.45 & 0.82 & 0.41 & 0.00 \\ 0.00 & 2.29 & 0.73 & 0.41 \\ 0.00 & 0.00 & 2.06 & 0.82 \\ 0.00 & 0.00 & 0.00 & 1.84 \end{bmatrix}$$



## 2.6 Esercizio 2.5.1

Consideriamo la matrice:

$$\mathbf{A} = \begin{bmatrix} -\epsilon & 1 \\ 1 & -1 \end{bmatrix}$$

Se  $\epsilon = 0$ , la fattorizzazione LU senza pivoting parziale fallisce per  $\mathbf{A}$ . Ma se  $\epsilon \neq 0$ , possiamo procedere senza pivoting, almeno in linea di principio.

- (a) Costruisci  $\mathbf{b} = \mathbf{Ax}$  prendendo  $\epsilon = 10^{-12}$  per la matrice  $\mathbf{A}$  e  $\mathbf{x} = [1, 1]$  come soluzione.
- (b) Fattorizza la matrice usando la decomposizione LU senza pivoting e risolvi numericamente per  $\mathbf{x}$ . Quanto è accurato il risultato?
- (c) Verifica la fattorizzazione LU calcolando  $\mathbf{A} - \mathbf{LU}$ . Funziona?
- (d) Ripeti per  $\epsilon = 10^{-20}$ . Quanto è accurato ora il risultato?
- (e) Calcola il numero di condizionamento della matrice  $\mathbf{A}$ ; puoi scegliere la norma che preferisci. Il sistema  $\mathbf{Ax} = \mathbf{b}$  è mal condizionato?
- (f) Calcola le matrici  $\mathbf{L}$  e  $\mathbf{U}$  della fattorizzazione LU di  $\mathbf{A} = \mathbf{LU}$  a mano.
- (g) Calcola i numeri di condizionamento di  $\mathbf{L}$  e  $\mathbf{U}$ . I sistemi triangolari  $\mathbf{Lz} = \mathbf{b}$  e  $\mathbf{Ux} = \mathbf{z}$  sono mal condizionati?

### 2.6.1 Soluzione

Per poter operare un confronto tra i risultati ottenuti con  $\epsilon = 10^{-12}$  e  $\epsilon = 10^{-20}$ , si è scelto di trattare il punto (d) insieme ai punti (a), (b) e (c). Il pedice 1 è legato al caso  $\epsilon = 10^{-12}$ , il pedice 2 al caso  $\epsilon = 10^{-20}$ .

- (a) Si è calcolato il prodotto tra la matrice  $\mathbf{A}$  e il vettore  $\mathbf{x}$ , ottenendo, a meno di arrotondamenti, i vettori:

$$\mathbf{b}_1 = \begin{bmatrix} 1.0 \\ 0.0 \end{bmatrix}, \quad \mathbf{b}_2 = \begin{bmatrix} 1.0 \\ 0.0 \end{bmatrix}. \quad (17)$$

- (b) Si è calcolata la fattorizzazione LU della matrice  $\mathbf{A}$ , ottenendo:

$$\mathbf{L}_1 = \begin{bmatrix} 1.0 & 0.0 \\ -1.0 \cdot 10^{12} & 1.0 \end{bmatrix}, \quad \mathbf{U}_1 = \begin{bmatrix} -1.0 \cdot 10^{-12} & 1.0 \\ 0.0 & 1.0 \cdot 10^{12} \end{bmatrix}, \quad (18)$$

$$\mathbf{L}_2 = \begin{bmatrix} 1.0 & 0.0 \\ -1.0 \cdot 10^{20} & 1.0 \end{bmatrix}, \quad \mathbf{U}_2 = \begin{bmatrix} -1.0 \cdot 10^{-20} & 1.0 \\ 0.0 & 1.0 \cdot 10^{20} \end{bmatrix}. \quad (19)$$

Si è risolto il sistema  $\mathbf{Ax} = \mathbf{b}$  utilizzando le sostituzioni triangolari, ottenendo:

$$\mathbf{x}_1 = \begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix}, \quad \mathbf{x}_2 = \begin{bmatrix} 0.0 \\ 1.0 \end{bmatrix}. \quad (20)$$

(c) Si è calcolato il prodotto tra le matrici  $\mathbf{L}$  e  $\mathbf{U}$ , in seguito lo si è sottratto ad  $\mathbf{A}$  ottenendo:

$$\mathbf{A}_1 - (\mathbf{L}\mathbf{U})_1 = \begin{bmatrix} 0.0 & 0.0 \\ 0.0 & 0.0 \end{bmatrix}, \quad \mathbf{A}_2 - (\mathbf{L}\mathbf{U})_2 = \begin{bmatrix} 0.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}. \quad (21)$$

È evidente che il risultato per  $\epsilon = 10^{-12}$  è accurato, al contrario del risultato ottenuto per  $\epsilon = 10^{-20}$ , in cui sia la soluzione che la verifica della fattorizzazione LU sono affette da errori numerici.

(e) Si è calcolato  $\kappa(A)$ , utilizzando sia la norma 1 che la norma infinito, ottenendo lo stesso numero a causa della natura simmetrica della matrice  $\mathbf{A}$ . La formula generale per il numero di condizionamento della matrice  $\mathbf{A}$  presa in esame è  $\kappa(A) = \frac{4}{1-\epsilon}$ .

$$\kappa_1(A) = \kappa_\infty(A) = 4.00000000000004, \quad \kappa_1(A) = \kappa_\infty(A) = 4.0. \quad (22)$$

Come si può notare, in entrambi i casi il numero di condizionamento non è molto grande, quindi il sistema  $\mathbf{Ax} = \mathbf{b}$  non è mal condizionato.

(f) Si sono calcolate a mano le matrici  $\mathbf{L}$  e  $\mathbf{U}$  della fattorizzazione LU di  $\mathbf{A}$  per ciascun valore di  $\epsilon$ . I calcoli dettagliati sono disponibili nella cartella `git lab\computazionale1\esercizi\pen_and_paper` il cui link è stato fornito in precedenza. Ad ogni modo le matrici ottenute sono già state riportate al punto 2.6.1.

(g) Si sono calcolati i numeri di condizionamento di  $\mathbf{L}$  e  $\mathbf{U}$ , utilizzando sia la norma 1 che la norma infinito, ancora una volta ottenendo lo stesso numero.

Le formule generali per il numero di condizionamento delle matrici  $\mathbf{L}$  e  $\mathbf{U}$  sono:

$$\kappa_1(L) = \kappa_\infty(L) = \left(1 + \frac{1}{\epsilon}\right)^2, \quad \kappa_1(U) = \kappa_\infty(U) = \frac{1}{\epsilon^2}. \quad (23)$$

In tabella 5 sono riassunti i risultati ottenuti.

Table 5: Valori dei parametri  $L$  e  $U$  per diverse tolleranze  $\epsilon$

	$\epsilon = 10^{-12}$	$\epsilon = 10^{-20}$
$L$	$1.0 \cdot 10^{24}$	$1.0 \cdot 10^{40}$
$U$	$1.0 \cdot 10^{24}$	$1.0 \cdot 10^{40}$

## 2.6.2 Conclusioni

L'esercizio ha mostrato come il numero di condizionamento di una matrice non sia un indicatore sufficiente per valutare l'esattezza della risoluzione del sistema lineare associato. Infatti, sebbene il numero di condizionamento della matrice  $\mathbf{A}$  sia relativamente basso, la risoluzione del sistema richiede il calcolo delle matrici  $\mathbf{L}$  e  $\mathbf{U}$ , il cui numero di condizionamento potrebbe risultare elevato. È questo il caso con  $\epsilon = 10^{-20}$ , in cui  $\kappa(A) = 4.0$  ma  $\kappa(L) = 1.0 \cdot 10^{40}$  e  $\kappa(U) = 1.0 \cdot 10^{40}$ . Dato che l'algoritmo utilizza le matrici  $\mathbf{L}$  e  $\mathbf{U}$  per risolvere il sistema è evidente che l'errore numerico aumenta a causa delle stesse.

## 2.7 Esercizio 2.6.1

- (a) Scrivi un programma che prenda in input una matrice  $\mathbf{A}$  e un vettore  $\mathbf{b}$  e risolva il problema dei minimi quadrati  $\arg\min \|\mathbf{b} - \mathbf{Ax}\|_2$  utilizzando la decomposizione di Cholesky.
- (b) Per verificare il funzionamento del tuo codice, calcola la soluzione ai minimi quadrati quando

$$\mathbf{A} = \begin{bmatrix} 2 & -1 \\ 0 & 1 \\ -2 & 2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ -5 \\ 6 \end{bmatrix}$$

### 2.7.1 Soluzione

Si è implementato l'algoritmo per il metodo dei minimi quadrati e lo si è testato sulle matrici proposte, ottenendo:

$$\mathbf{x} = \begin{bmatrix} -2.0 \\ -1.0 \end{bmatrix}. \quad (24)$$

## 2.8 Esercizio 2.6.2

Keplero ha scoperto che il periodo orbitale  $\tau$  di un pianeta dipende dalla sua distanza media  $R$  dal Sole secondo la legge  $\tau = cR^\alpha$ , dove  $\alpha$  è un semplice numero razionale.

- (a) Esegui un fit lineare ai minimi quadrati utilizzando la tabella 8 per determinare il valore più probabile e semplice di  $\alpha$ .

Il risultato è in accordo con le tue aspettative? [Suggerimento: ricorda le leggi del moto planetario di Keplero.]

- (b) Realizza un grafico dei dati e del risultato del fit.

### 2.8.1 Soluzione

(a) Il metodo dei minimi quadrati che è stato sviluppato durante il corso è adatto a risolvere problemi lineari, in cui le costanti che si vogliono determinare sono coefficienti lineari di funzioni dei dati. Non è il caso del problema proposto, dato che  $\alpha$  si trova ad esponente. Per risolvere la questione si è scelto di linearizzare il problema tramite il logaritmo naturale. In formule:

$$\ln(\tau) = \ln(c) + \alpha \ln(R). \quad (25)$$

Da qui in poi si è implementato il metodo dei minimi quadrati, applicando la scomposizione di Cholesky alla matrice  $n \times n$  ottenuta dal prodotto della matrice di design e la sua trasposta.

Per chiarezza si riporta la matrice di design:

$$\mathbf{A} = \begin{bmatrix} 1 & \ln(57.59) \\ 1 & \ln(108.11) \\ 1 & \ln(149.57) \\ 1 & \ln(227.84) \\ 1 & \ln(778.14) \\ 1 & \ln(1427) \\ 1 & \ln(2870.3) \\ 1 & \ln(4499.9) \end{bmatrix}. \quad (26)$$

La matrice di regressione è stata costruita in modo che la prima colonna fosse composta da 1, per poter calcolare il termine noto della retta, e la seconda colonna fosse composta dai logaritmi naturali delle distanze dei pianeti dal Sole.

La legge di Keplero prevista è la seguente:

$$\tau = cR^{3/2}, \quad (27)$$

dove  $\alpha = 3/2$ .

È necessario un ulteriore commento su 27, prima di riportare i risultati. Fino ad ora abbiamo assunto che la costante  $c$  fosse indipendente dal pianeta, ma in realtà la sua espressione è:

$$c = \frac{2\pi}{\sqrt{G(M+m)}}, \quad (28)$$

dove  $G$  è la costante di gravitazione universale,  $M$  è la massa del Sole,  $m$  è la massa del pianeta.

Consideriamo Giove, il pianeta più massivo di quelli presi in esame ( $m \approx 1.90 \cdot 10^{27}$  kg). Si ha:

$$c_1 = \frac{2\pi}{\sqrt{G(M+m)}} = 5.47 \cdot 10^{-10} \frac{\text{sec}}{m^{3/2}} = 0.203 \frac{\text{days}}{Mkm^{3/2}} \quad (29)$$

$$c_2 = \frac{2\pi}{\sqrt{GM}} = 5.47 \cdot 10^{-10} \frac{\text{sec}}{m^{3/2}} = 0.203 \frac{\text{days}}{Mkm^{3/2}} \quad (30)$$

$$\frac{c_1}{c_2} \approx 1.0 \quad (31)$$

Possiamo quindi affermare che la costante  $c$  è indipendente dalla massa del pianeta, entro la precisione necessaria per il nostro scopo.

Si è quindi calcolato il valore di  $\alpha$  e della costante  $c$ , ottenendo:

$$c = 0.206 \frac{\text{days}}{Mkm^{3/2}} \quad \alpha = 1.499 \quad (32)$$

I risultati sono in accordo con quanto atteso.

**(b)** Si riporta in figura 9 il grafico dei dati e relativo fit.

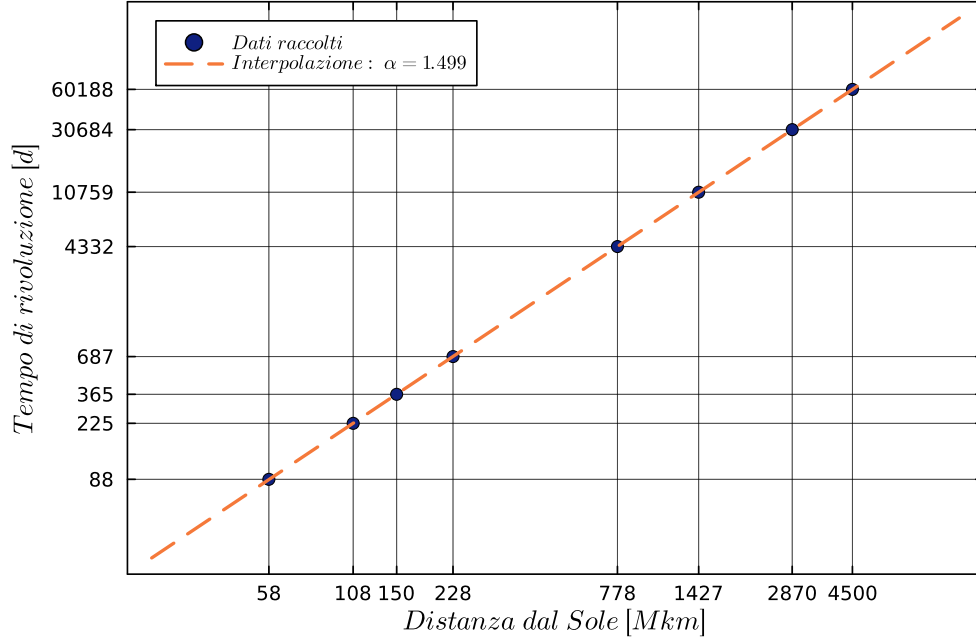


Figure 9: Fit lineare ai minimi quadrati dei dati dei pianeti del Sistema Solare.

## 2.9 Esercizio 2.6.3

In questo esercizio si vuole trovare un'approssimazione della funzione periodica  $g(t) = e^{\sin(t-1)}$  su un periodo,  $0 < t \leq 2\pi$ . Come dati, si definiscono

$$t_i = \frac{2\pi i}{60}, \quad y_i = g(t_i), \quad i = 1, \dots, 60. \quad (33)$$

(a) Trova i coefficienti del fit ai minimi quadrati

$$y(t) \approx c_1 + c_2 t + \dots + c_7 t^6. \quad (34)$$

Sovrapponi un grafico dei valori dei dati come punti con una curva che mostra il fit.

(b) Trova i coefficienti del fit ai minimi quadrati

$$y \approx d_1 + d_2 \cos(t) + d_3 \sin(t) + d_4 \cos(2t) + d_5 \sin(2t). \quad (35)$$

A differenza del punto (a), questa funzione di fitting è essa stessa periodica. Sovrapponi un grafico dei valori dei dati come punti con una curva che mostra il fit.

### 2.9.1 Soluzione

Dopo avere generato i dati richiesti, si è eseguito il metodo dei minimi quadrati con le seguenti matrici di regressione:

$$\mathbf{D}_{\text{pol}} = \begin{bmatrix} 1 & t_1 & \dots & t_1^6 \\ 1 & t_2 & \dots & t_2^6 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & t_{60} & \dots & t_{60}^6 \end{bmatrix}, \quad \mathbf{D}_{\text{Four}} = \begin{bmatrix} 1 & \cos(t_1) & \sin(t_1) & \cos(2t_1) & \sin(2t_1) \\ 1 & \cos(t_2) & \sin(t_2) & \cos(2t_2) & \sin(2t_2) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \cos(t_{60}) & \sin(t_{60}) & \cos(2t_{60}) & \sin(2t_{60}) \end{bmatrix}, \quad (36)$$

ottenendo i coefficienti delle formule 34 e 35:

- Coefficienti del polinomio:  $[0.712; -1.351; 2.225; -0.538; -0.074; 0.033; -0.003]$
- Coefficienti della funzione di Fourier:  $[1.266; -0.951; 0.611; 0.113; -0.247]$

Si riportano i grafici dei dati e dei fit in figura 10.

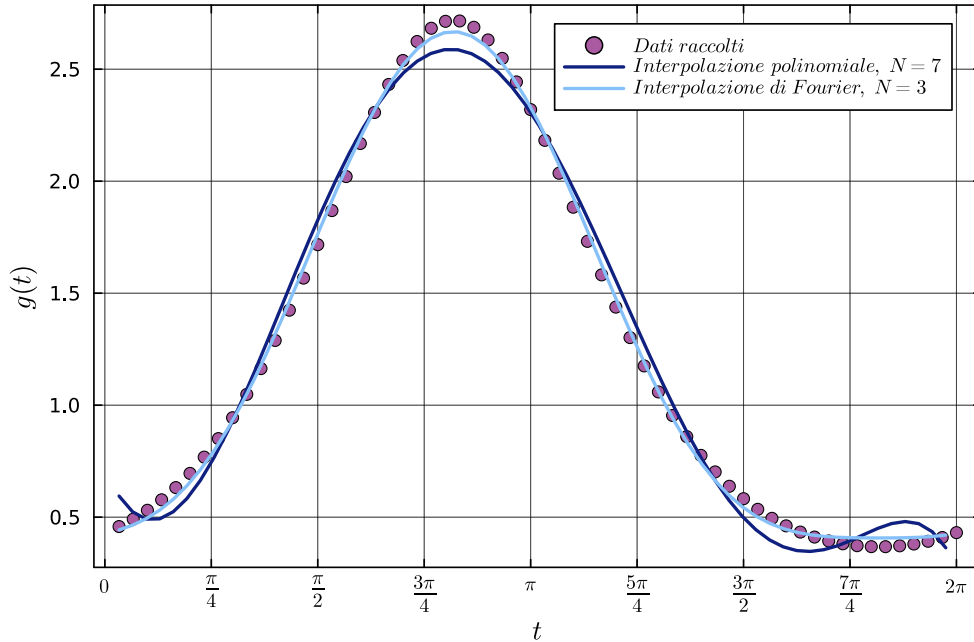


Figure 10: Fit polinomiale e di Fourier ai minimi quadrati della funzione  $g(t) = e^{\sin(t-1)}$ .

### 2.9.2 Conclusioni

In figura 10 si osserva che il fit meglio riuscito è quello di Fourier, nonostante richieda meno parametri. Il motivo è che la funzione  $g(t)$  è periodica, e quindi il fit con il modello 35 riesce a catturare meglio la periodicità della funzione rispetto al fit eseguito con 34.

In un contesto sperimentale la funzione  $g(t)$  non è nota a priori, anzi lo scopo dell'esperimento è solitamente quello di risalire a tale funzione. Il fatto che il fit secondo il modello di Fourier riesca meglio è indice della natura periodica del fenomeno osservato.

## 3 Radici di equazioni non lineari

### 3.1 Teoria

### 3.2 Esercizi 3.2.1, 3.2.2

1. Implementa il metodo di bisezione e testalo con una funzione semplice, ad esempio  $f(x) = x$ .

2. (a) Usa il metodo di bisezione per trovare le soluzioni di  $x^3 - 7x^2 + 14x - 6 = 0$  su  $[0, 1]$ .  
(b) Studia la convergenza del metodo di bisezione rispetto alla radice trovata in (a). (Puoi prendere come approssimazione per la radice  $r = m_{n_{\max}}$ , dove  $n_{\max}$  è l'indice corrispondente all'ultima iterazione di bisezione.)

Qual è il suo ordine di convergenza  $q$ ? Puoi stimare la costante d'errore asintotica  $C$ ?

Suggerimento: Studia  $d_n = -\log_{10} |x_n - r|$  come funzione di  $n$ .

#### 3.2.1 Soluzione

(1) Si è implementato il metodo di bisezione, e si è testato con la funzione  $f(x) = x$  per due intervalli:  $[-1.0, 1.0]$  e  $[-5.0, 1.0]$ . Il motivo della scelta di questi due intervalli è che ci aspettiamo che il metodo restituisca  $x = 0$  come soluzione e nel caso di intervallo simmetrico il numero di iterazioni è pari ad 1. Con un intervallo asimmetrico il test è più efficiente perchè il numero di iterazioni aumenta.

Il metodo si interrompe quando la distanza tra gli estremi risulta minore di  $\max(|a|, |b|) \cdot 10^{-16}$ , dove  $a$  e  $b$  sono gli estremi dell'intervallo. Il metodo ha prodotto i seguenti risultati:

- Intervallo  $[-1.0, 1.0]$ :  $x = 0.0$  in 1 iterazione.
- Intervallo  $[-5.0, 1.0]$ :  $x = 1.1 \cdot 10^{-162}$  in 539 iterazioni.

(2a) Si è implementato il metodo di bisezione per la funzione  $f(x) = x^3 - 7x^2 + 14x - 6$ . Il metodo ha restituito un valore della radice  $r = 0.5857864376269051$  con  $f(r) = 0.0$ , in 50 iterazioni. In figura 11 si riporta il grafico della funzione e delle radici successive trovate.

(2b) Per studiare la convergenza del metodo di bisezione bisogna fare alcune considerazioni preliminari.

È noto che per il metodo di bisezione vale la relazione  $|m_n - r| < 2^{-(n+1)}(b_0 - a_0)$ , dove  $m_n$  è la stima della radice al passo  $n$ ,  $r$  è la radice cercata,  $a_0$  e  $b_0$  sono gli estremi dell'intervallo iniziale.

Ad ogni iterazione l'intervallo viene dimezzato, in modo che al passo successivo la stima della radice si trovi in una delle due metà dell'intervallo nella quale si trovava la stima precedente. In formule:  $|m_{n+1} - r| < \frac{|m_n - r|}{2}$ .

Unendo le due relazioni presentate sopra si ricava  $\frac{|m_{n+1} - r|}{|m_n - r|} < \frac{1}{2}$ , e passando al limite:

$$\lim_{n \rightarrow \infty} \frac{|m_{n+1} - r|}{|m_n - r|} = \frac{1}{2}, \quad (37)$$

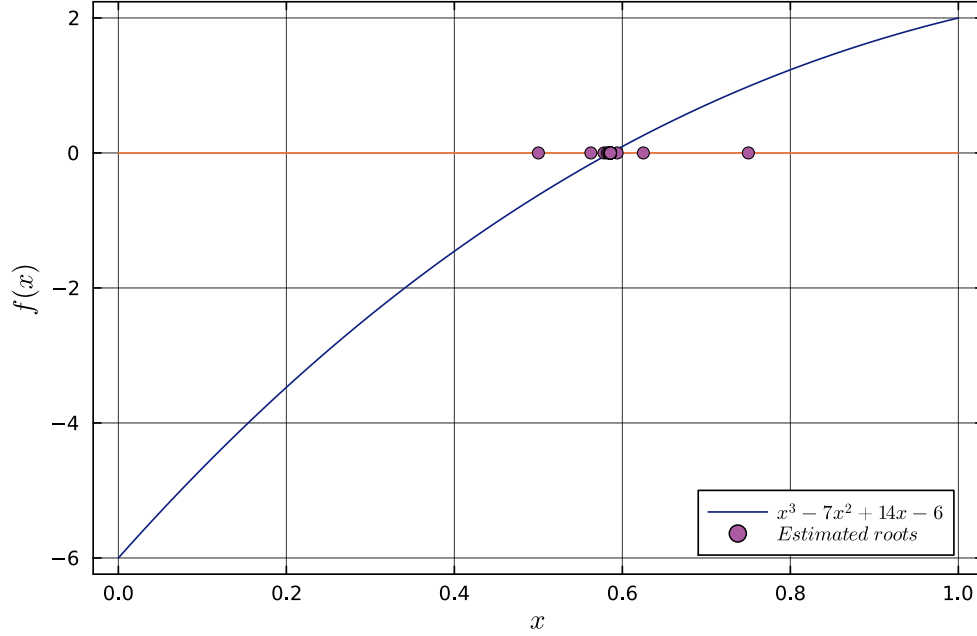


Figure 11: Grafico della funzione  $f(x) = x^3 - 7x^2 + 14x - 6$  e della radice trovata con il metodo di bisezione.

da cui si ottiene che  $q$ , ordine di convergenza, è 1, mentre  $C$ , la costante di errore asintotico, vale  $\frac{1}{2}$ .

Verifichiamolo numericamente, ricordando le relazioni:

Per verificare i valori teorici di  $q$  e  $C$  è necessario tenere a mente le seguenti relazioni. La prima si ottiene applicando il logaritmo decimale alla 37:

$$d_{n+1} \approx q d_n - \log_{10} |C| \quad \Rightarrow \quad \frac{d_{n+1}}{d_n} \approx q - \frac{\log_{10} |C|}{d_n} \xrightarrow{n \rightarrow \infty} q \quad (38)$$

dove  $d_n = -\log_{10} |m_n - r|$ .

La seconda si ottiene iterando la relazione  $|x_{k+1} - r| = C|x_k - r|$ :

$$|x_{k+1} - r| = C^{k+1} |x_0 - r| \quad (39)$$

**Stima di  $q$**  Per stimare  $q$  si è calcolato il rapporto tra i valori di  $d_n$  e  $d_{n+1}$  e lo si è messo in grafico in funzione di  $n$ , ottenendo la figura 12, in cui è presente un ingrandimento per i valori più alti di  $n$ .

Il rapporto tende a 1, come atteso. Si vuole ora quantificare tale valore.

La relazione 38 vale per  $n \rightarrow \infty$ . Ciò significa che la stima di  $q$  dovrebbe essere calcolata per  $n$  più grande possibile. Ciononostante si è deciso di stimare il valore di  $q$  tramite l'interpolazione con funzione costante perchè il valore di  $q$  fluttua attorno a 1, ed è possibile che per  $n$  maggiori di quelli disponibili il valore cercato continui a fluttuare attorno a 1. Per l'interpolazione si sono considerati gli ultimi 20 punti della serie, riportati nell'ingrandimento di 12. Il fit su tali punti restituisce un valore di  $q = 1.02$ .



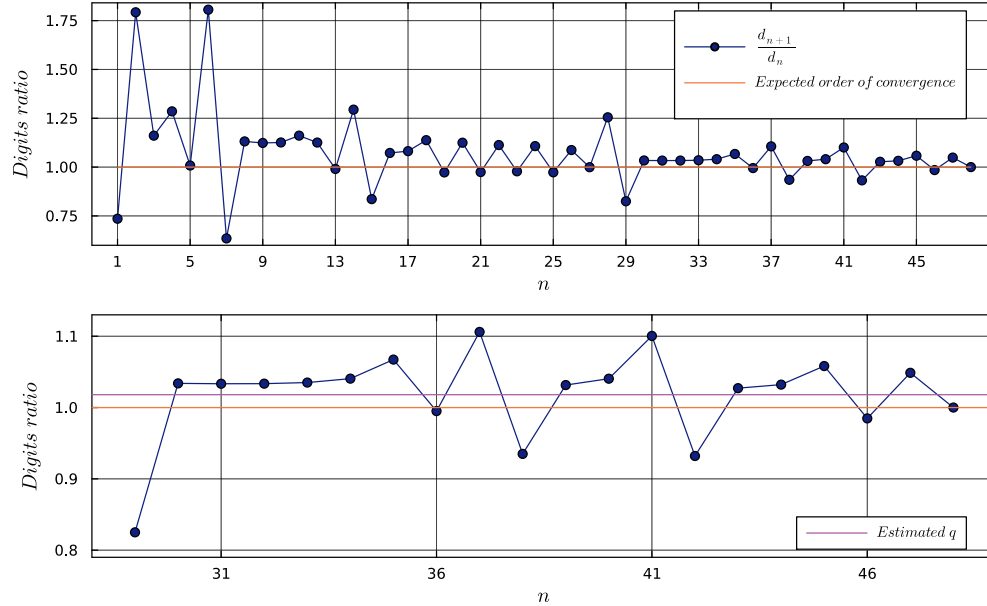


Figure 12: Rapporto tra i valori di  $d_n$  e  $d_{n+1}$  in funzione di  $n$ .

**Stima di  $C$**  Per stimare  $C$  si è eseguito il fit con metodo dei minimi quadrati sul modello:

$$\log|x_{k+1} - r| = (k + 1)\log(C) + \log|x_0 - r| \rightarrow y(x) = x\log(C) + A, \quad (40)$$

avendo considerato  $r$  come l'ultimo valore di  $m_n$  calcolato. Si riporta il grafico del fit in figura 13. Il fit ha restituito i seguenti valori:

$$C = 0.5, \quad A = -0.1. \quad (41)$$

Si osserva che il valore di  $C$  è in accordo con quanto atteso, mentre il valore di  $A$  non è significativo.

**Commento:** È possibile stimare i valori di  $q$  e  $C$  anche in modo diverso, e con un solo fit. Interpolando la relazione 38 con un'iperbole, e ignorando il limite, si riescono a ottenere i valori di  $q$  e  $C$  in un colpo solo. I risultati ottenuti hanno una precisione minore di quelli ottenuti con i metodi precedenti, pertanto non sono riportati. Si rimanda alla cartella `git lab\computazionale1\esercizi\3_2_2.ipynb` per i dettagli.

### 3.3 Esercizi 3.3.1, 3.3.2

1. Scrivi un programma che implementi il metodo di Newton.
2. Per ciascuna delle seguenti funzioni, esegui i seguenti passi.
  - (a) Riscrivi l'equazione nella forma standard per la ricerca degli zeri,  $f(x) = 0$ , e calcola  $f'(x)$ .
  - (b) Traccia il grafico di  $f$  nell'intervallo indicato e determina quante radici sono presenti

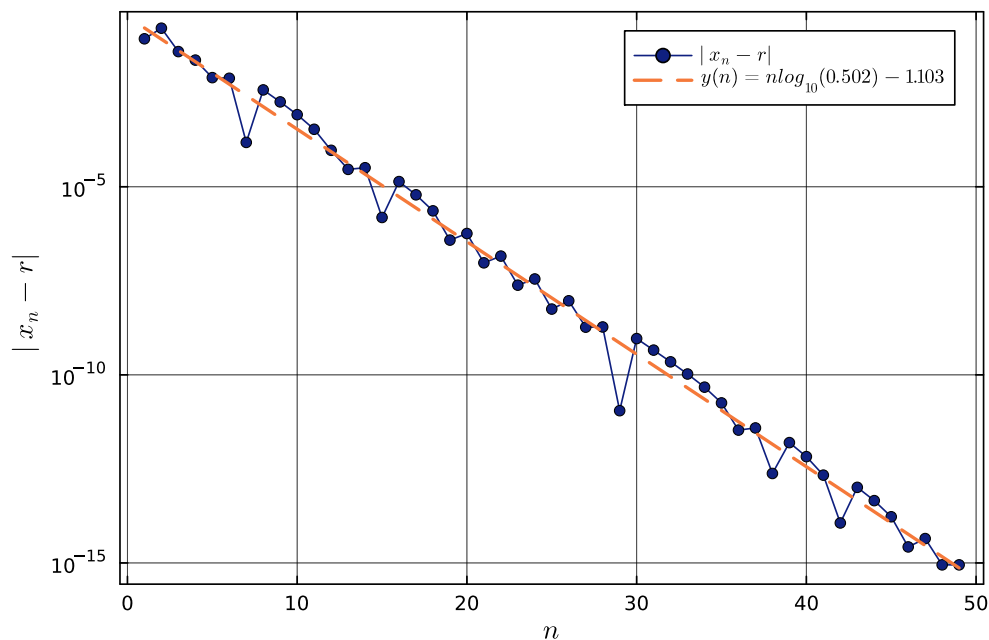


Figure 13: Fit lineare del logaritmo decimale della distanza tra la radice e le stime successive.

nell'intervallo.

(c) Usa il metodo di Newton per trovare ciascuna radice.

(d) Studia l'errore nella sequenza di Newton per determinare numericamente se la convergenza è approssimativamente quadratica per la radice trovata.

Suggerimento: Studia  $d_{n+1}/d_n$  con  $d_n = -\log_{10} |x_n - r|$  per verificare il tasso di convergenza  $q$ . Una volta stabilito, prova a stimare la costante d'errore asintotica  $C$ .

- $x^2 = e^{-x}$ , su  $[-2, 2]$
- $2x = \tan x$ , su  $[-0.2, 1.4]$
- $e^{x+1} = 2 + x$ , su  $[-2, 2]$

### 3.3.1 Soluzione

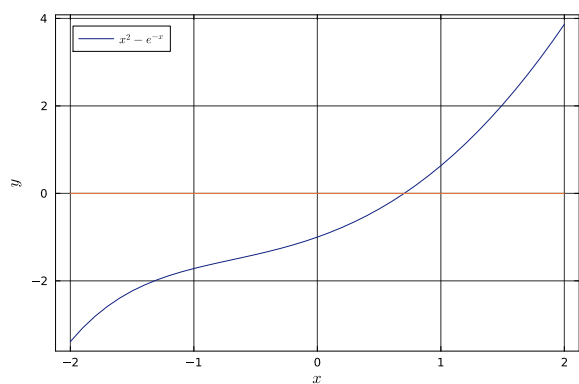
(1) Si è implementato il metodo di Newton, e si è testato con la funzione  $f(x) = e^x - 1$  con valore iniziale  $x_0 = 10$ . Il metodo ha restituito il valore  $-6.68 \cdot 10^{-17}$  in 16 iterazioni. Valore atteso:  $x = 0$ .

(2a) Le funzioni in forma di ricerca degli zeri sono:

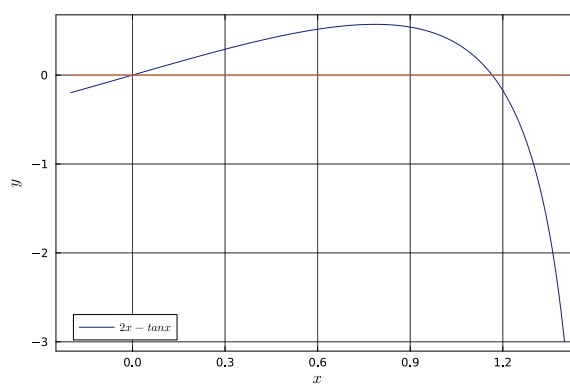
- $f_1(x) = x^2 - e^{-x}$ , con  $f'_1(x) = 2x + e^{-x}$ , su  $[-2, 2]$ .
- $f_2(x) = 2x - \tan x$ , con  $f'_2(x) = 2 - \tan^2 x$ , su  $[-0.2, 1.4]$ .
- $f_3(x) = e^{x+1} - 2 - x$ , con  $f'_3(x) = e^{x+1} - 1$ , su  $[-2, 2]$ .

(2b) I grafici delle funzioni sono riportati in figura 14, da cui si evince che:

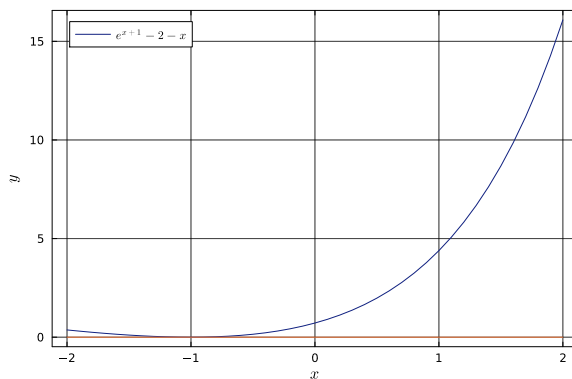
- $f_1$  ha una radice nell'intervallo  $[-2, 2]$ .
- $f_2$  ha due radici nell'intervallo  $[-0.2, 1.4]$ .
- $f_3$  ha una radice nell'intervallo  $[-2, 2]$ .



(a)



(b)



(c)

Figure 14: Grafici delle funzioni  $f_1$ ,  $f_2$  e  $f_3$ .

(2c) L'algoritmo di Newton è stato applicato alle tre funzioni, con i seguenti risultati:

- $f_1$ : radice  $x = 0.703$ , in 8 iterazioni. Punto di partenza  $x_0 = -2$ .

- $f_2$ : radici  $x_1 = 0.000$ ,  $x_2 = 1.166$ , entrambe in 7 iterazioni. Punti di partenza  $x_{0,1} = 0.65$ ,  $x_{0,2} = 1.4$ .
- $f_3$ : radice  $x = -1.000$  in 29 iterazioni. Punto di partenza  $x_0 = 2$ .

**(2d) - Prima funzione** Scopo dello studio dell'errore è verificare se i valori di  $q$  e  $C$  sono in accordo con quanto atteso. In particolare ci aspettiamo che il metodo di Newton per questo particolare zero abbia  $q = 2$  e  $C = \frac{1}{2} \left| \frac{f''(r)}{f'(r)} \right| = 0.396$ . Per determinare i valori di  $q$ , qui e nelle altre funzioni, si fa riferimento alla formula 38, e si calcola il rapporto tra i valori di  $d_n$  e  $d_{n+1}$ ; per determinare  $C$ , qui e nelle altre funzioni, si fa riferimento alla relazione  $\lim_{n \rightarrow \infty} \frac{|x_{n+1}-r|}{|x_n-r|^q} = \lim_{n \rightarrow \infty} \frac{|\epsilon_{n+1}|}{|\epsilon_n|^q} = C$ . Si riportano i due grafici in figura 15.

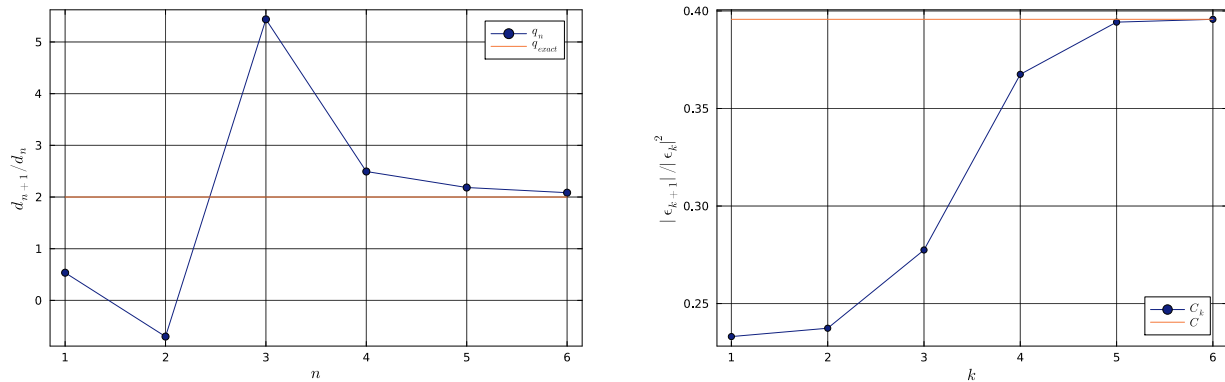


Figure 15: Studio del valore di  $q$  e  $C$  per  $f_1$ .

Per le stime di  $q$  e  $C$  si è scelto di considerare l'ultimo punto della successione corrispondente, a causa del numero limitato di iterazioni. Si ottiene:

$$q = 2.084, \quad C = 0.396. \quad (42)$$

Si osserva che i valori ottenuti sono in accordo con quanto atteso, e quindi si può concludere che il metodo di Newton converge in modo quadratico per la radice trovata.

## (2d) - Seconda funzione

**Prima radice,  $x = 0.0$ :** È importante notare che la funzione  $f_2$  in zero ha un andamento lineare, e perciò la sua derivata seconda è nulla. Questo implica che i valori di  $q$  e  $C$  non sono determinati dalle formule precedenti, perchè tali formule sono ricavate a partire da un'espansione di Taylor al secondo ordine. Per ovviare al problema si è scelto di ricavare le formule già utilizzate ma a ordine 3. I calcoli sono riportati nella cartella `git lab\computazionale1\esercizi\pen_and_paper`.

Di seguito sono riportati i valori per  $q$  e  $C$  con le dovute correzioni:

$$q = 3, \quad C = \frac{f'''(r)}{3f'(r)} = 0.667. \quad (43)$$

In figura 16 si riportano i grafici delle successioni  $q_n$  e  $C_n$  per la prima radice di  $f_2$ .

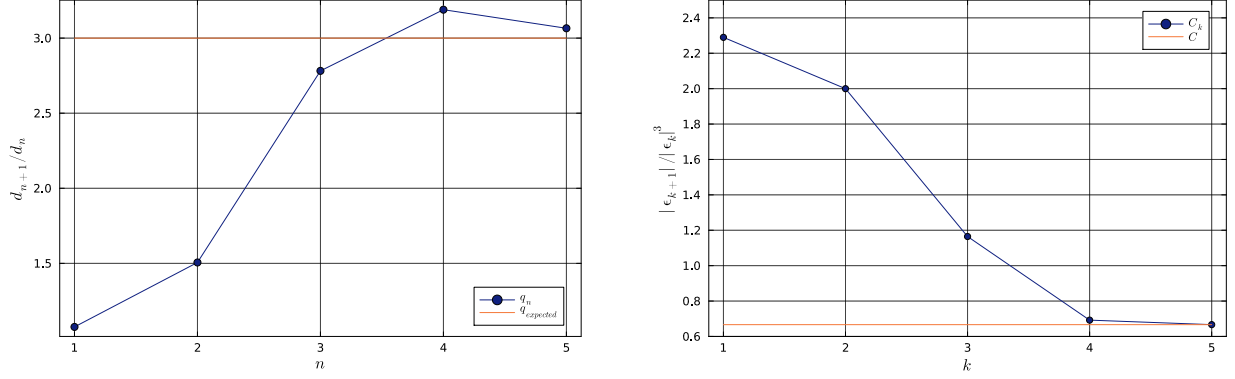


Figure 16: Studio del valore di  $q$  e  $C$  per  $f_2$ , prima radice.

Si ottengono i seguenti valori:

$$q = 3.065, \quad C = 0.667. \quad (44)$$

Ci si può chiedere cosa succede se, non conoscendo l'andamento della funzione, si utilizzano le formule precedenti. In particolare cosa succede se si fissa  $q = 2$  e si stima  $C$  come l'ultimo elemento della successione  $C_n = \frac{|x_{n+1}-r|}{|x_n-r|^2}$ . L'aspettativa è che  $C$  risulti nullo, dato che la derivata seconda è nulla. Il grafico corrispondente può essere trovato nella cartella git `lab\computazionale1\esercizi\3_3_2.ipynb`, e mostra che il valore di  $C$  è effettivamente nullo.

**Seconda radice,  $x = 1.166$ :** la seconda radice di  $f_2$  non presenta problemi, così le formule sono quelle viste inizialmente. I valori attesi sono  $q = 2$  e  $C = 3.383$ . Si riportano i grafici delle successioni  $q_n$  e  $C_n$  in figura 17. Si ottengono i seguenti valori:

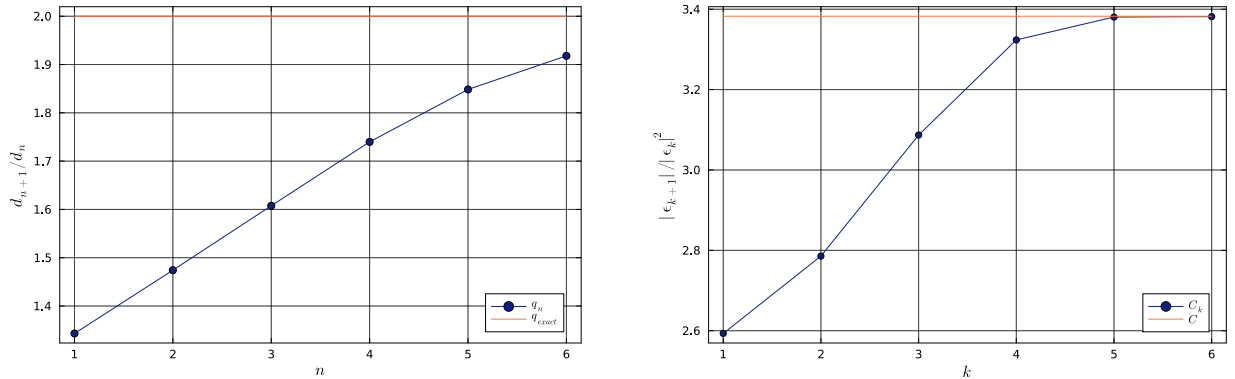


Figure 17: Studio del valore di  $q$  e  $C$  per  $f_2$ , seconda radice.

$$q = 2.000, \quad C = \frac{f''(r)}{2f'(r)} = 3.382. \quad (45)$$

In accordo con quanto atteso, il metodo di Newton converge in modo quadratico per la radice trovata.

**(2d) - Terza funzione** Osservando il grafico della funzione  $f_3$  si nota che la radice è un punto di minimo, e quindi ha una molteplicità  $m$ . D'altra parte, possiamo intuire che la radice di  $f_3$  ha una molteplicità anche dal numero di iterazioni, molto elevato rispetto alle altre funzioni.

Come è noto dalla teoria, il metodo di Newton in caso di radici multiple converge linearmente, e nella formula  $\epsilon_{n+1} = C(\epsilon_n)^q$  ci aspettiamo  $q = 1$ . Inoltre in caso di radici multiple la costante d'errore asintotica diviene  $(1 - \frac{1}{m})$ .

Il primo passo è verificare il valore di  $q$ . Si riporta in figura 18 il grafico della successione  $q_n$ , da cui possiamo confermare stimare  $q = 1.067$ .

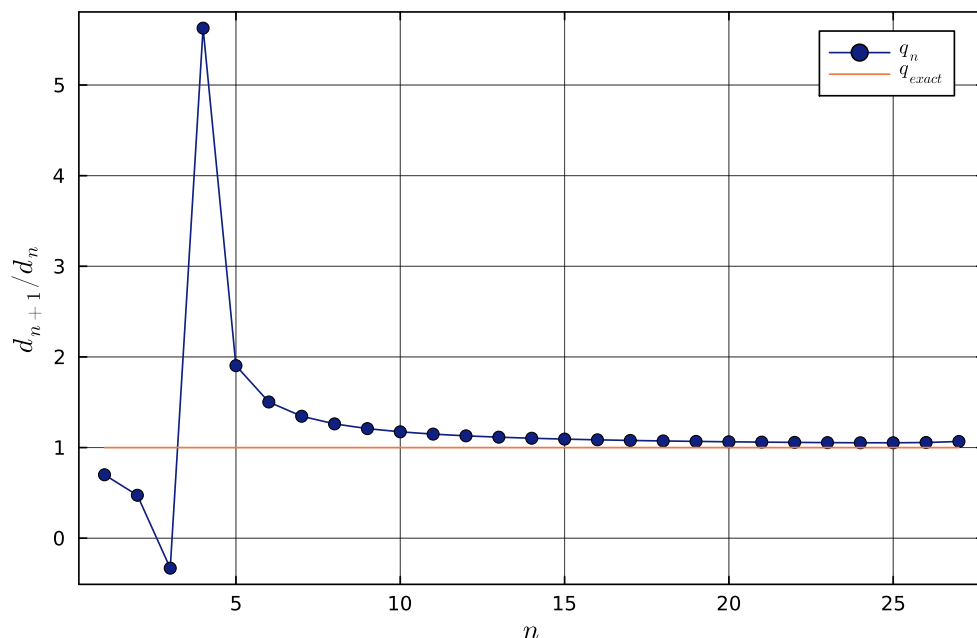


Figure 18: Studio del valore di  $q$  per  $f_3$ .

Si riportano in figura 19 i grafici della successione  $C_n$ , una volta senza considerare la molteplicità della radice, e una volta considerando la molteplicità della radice.

Come si può notare da 19, non avendo considerato la molteplicità della radice, il valore di  $C$  si discosta molto da quello calcolato con la solita formula  $C = \frac{f''(r)}{2f'(r)}$ . Invece, nel secondo grafico della stessa figura si osserva che i valori successivi di  $C_n$  si stabilizzano attorno al valore  $C = 0.5$ , per poi calare nuovamente a causa di errori di cancellazione numerica. Scegliendo  $C = C_{15}$  come stima del plateau si ottengono i seguenti valori:

$$C = 0.500, \quad m = 2.000, \quad (46)$$

in accordo con quanto atteso.

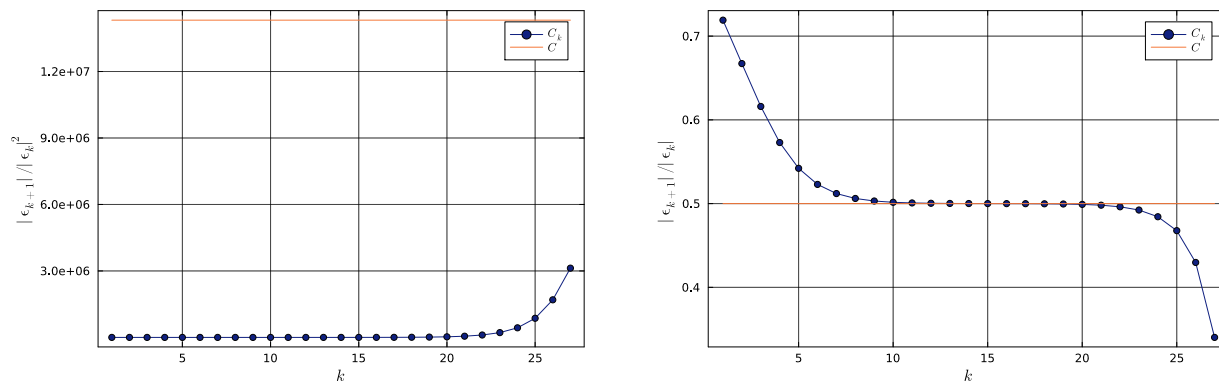


Figure 19: Studio del valore di  $C$  per  $f_3$ .

### 3.4 Esercizio 3.3.3

Traccia la funzione  $f(x) = x^{-2} - \sin x$  nell'intervallo  $x \in [0.5, 10]$ . Per ciascun valore iniziale  $x_1 = 1, x_1 = 2, \dots, x_1 = 7$ , applica il metodo di Newton a  $f$  e realizza una tabella che mostri  $x_1$  e la radice trovata dal metodo. In quale caso l'iterazione converge a una radice diversa da quella più vicina al valore iniziale? Usa il grafico per spiegare perché ciò è accaduto.

#### 3.4.1 Soluzione

Si riporta in figura 20 il grafico della funzione  $f(x) = x^{-2} - \sin x$  nell'intervallo  $[0.5, 10]$ , con i punti di partenza del metodo di Newton.

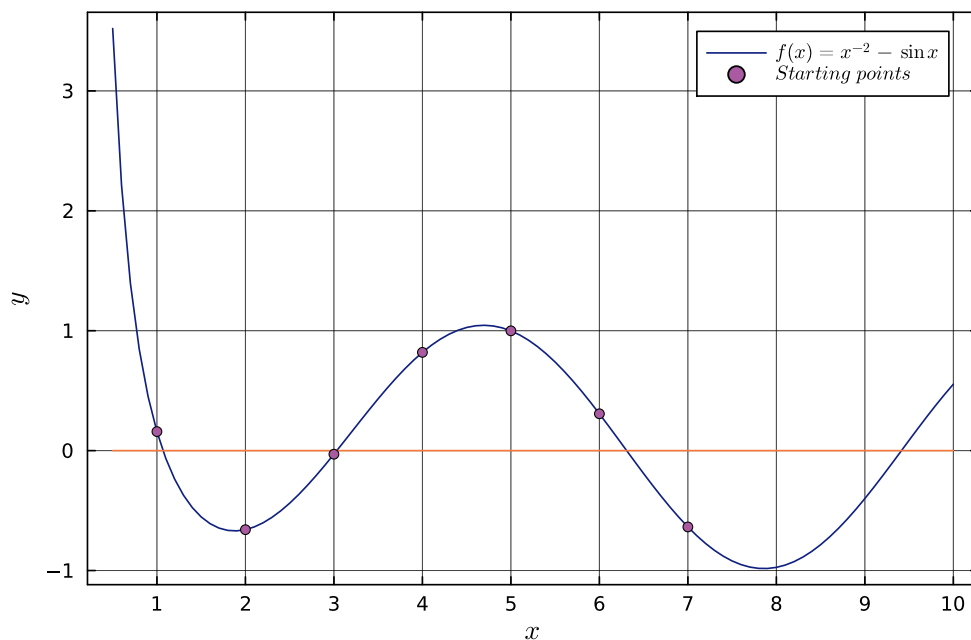


Figure 20: Grafico della funzione  $f(x) = x^{-2} - \sin x$  nell'intervallo  $[0.5, 10]$ .

Applicando il metodo di Newton con i valori iniziali, non sempre si ottengono le radici

più vicine al valore di partenza.  
Si riportano i risultati in tabella 6.

Table 6: Valori relativi ai punti  $x_1, \dots, x_7$

	$x_1 = 1$	$x_2 = 2$	$x_3 = 3$	$x_4 = 4$	$x_5 = 5$	$x_6 = 6$	$x_7 = 7$
$x_{\text{start}}$	1.0	2.0	3.0	4.0	5.0	6.0	7.0
$x_{\text{root}}$	1.1	6.3	3.0	3.0	9.4	6.3	6.3
$f'(x)$	-2.5	0.2	0.9	0.6	-0.3	-0.9	-0.8

Le due colonne evidenziate in arancione sono i punti iniziali che non convergono alla radice più vicina. Osservando il grafico della funzione è facile osservare che i punti iniziali  $x_2$  e  $x_5$  sono vicini a un minimo e a un massimo locali, rispettivamente. In effetti la derivata prima in questi punti è vicina a zero, come riportato in tabella. Non è esattamente zero, altrimenti il caso sarebbe quello della seconda funzione all'esercizio 3.3.2, ma è comunque piccola: ciò significa che il metodo di Newton converge quadraticamente, ma non necessariamente alla radice più vicina.

### 3.5 Esercizio 3.3.4

Le proprietà più facilmente osservabili dell'orbita di un corpo celeste attorno al Sole sono il periodo  $\tau$  e l'eccentricità ellittica  $\epsilon$ . (Un cerchio ha  $\epsilon = 0$ .) Da questi parametri è possibile determinare, in ogni istante  $t$ , la *vera anomalia*  $\theta(t)$ . Questa è l'angolo tra la direzione del perielio (il punto dell'orbita più vicino al Sole) e la posizione attuale del corpo, visto dal fuoco principale dell'ellisse (dove si trova il Sole). Questo si ottiene tramite

$$\tan \frac{\theta}{2} = \sqrt{\frac{1+\epsilon}{1-\epsilon}} \tan \frac{\psi}{2} \quad (47)$$

dove l'*anomalia eccentrica*  $\psi(t)$  soddisfa l'equazione di Keplero:

$$\psi - \epsilon \sin \psi - \frac{2\pi t}{\tau} = 0. \quad (48)$$

L'equazione 48 deve essere risolta numericamente per trovare  $\psi(t)$ , e poi la 47 può essere risolta analiticamente per ottenere  $\theta(t)$ . L'asteroide Eros ha  $\tau = 1.7610$  anni e  $\epsilon = 0.2230$ . Utilizzando il metodo di Newton per la 48, realizza un grafico di  $\theta(t)$  per 100 valori di  $t$  compresi tra 0 e  $\tau$ , cioè per un'orbita completa.

(Nota: usa  $\text{mod}(\theta, 2\pi)$  per riportare l'angolo tra 0 e  $2\pi$  se vuoi che il risultato sia una funzione continua.)

### 3.6 Soluzione

Per prima cosa si è risolta l'equazione 48 con il metodo di Newton. Per capire meglio il procedimento si riporta l'equazione 48 nel grafico 21, dove si osserva che la funzione ha un solo zero nell'intervallo  $[0, 2\pi]$ . Il metodo di Newton è stato applicato al variare del tempo  $t$ ,



cioè è stato applicato alla stessa funzione ma traslata in verticale, come si può vedere nella figura.

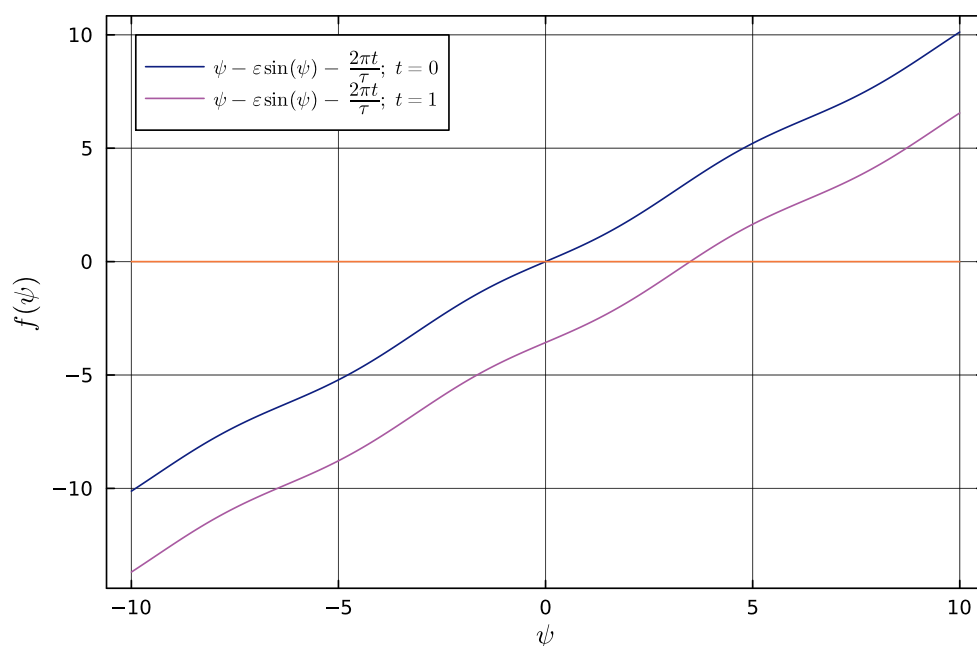


Figure 21: Grafico di  $\psi$  in corrispondenza di due tempi.

Ottenuti i valori di  $\psi$  in corrispondenza dei 100 valori di  $t$  compresi tra 0 e  $\tau$ , si è calcolata la vera anomalia  $\theta$  tramite la relazione 47. Si riporta il grafico di  $\theta(t)$  in figura 22

### 3.7 Esercizi 3.4.1, 3.4.2

1. Scrivi un programma che implementi il metodo della secante.
2. Per ciascuna delle seguenti funzioni, esegui i seguenti passi.
  - (a) Riscrivi l'equazione nella forma standard per la ricerca degli zeri,  $f(x) = 0$ .
  - (b) Traccia il grafico di  $f$  nell'intervallo indicato e determina quante radici sono presenti nell'intervallo.
  - (c) Per ciascuna radice determina un intervallo che la racchiuda. Poi usa il metodo della secante, partendo dagli estremi dell'intervallo, per trovare ciascuna radice.
  - (d) Per una delle radici, usa gli errori nella sequenza della secante per determinare numericamente se la convergenza è apparentemente compresa tra lineare e quadratica.
1.  $x^2 = e^{-x}$ , su  $[-2, 2]$
2.  $2x = \tan x$ , su  $[-0.2, 1.4]$
3.  $e^{x+1} = 2 + x$ , su  $[-2, 2]$

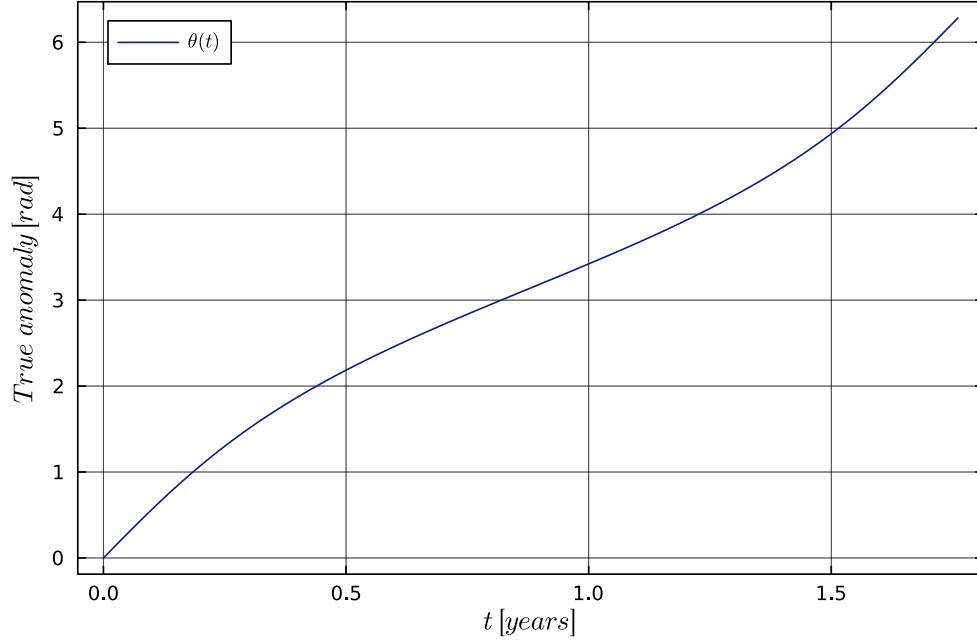


Figure 22: Grafico della vera anomalia  $\theta(t)$  per l'asteroide Eros.

### 3.7.1 Soluzione

(1) Si è implementato il metodo della secante e lo si è testato con la funzione  $x^2 - 1$ , ottenendo come radice  $x = 1.0$  in 10 iterazioni.

(2a)-(2b) Per questi esercizi si rimanda alle sezioni 3.3.1 e 3.3.1.

(2c) L'algoritmo della secante è stato applicato alle tre funzioni, con i seguenti risultati:

- $f_1$ : radice  $x = 0.703$ , in 9 iterazioni. Intervallo iniziale  $[1, 2]$ .
- $f_2$ : radici  $x_1 = 0.000$ ,  $x_2 = 1.166$ , in 6 e 11 iterazioni, rispettivamente. Intervalli iniziali:  $[-0.1, 0.2]$  e  $[0.9, 1.3]$ .
- $f_3$ : radice  $x = -1.000$  in 42 iterazioni. Intervallo iniziale  $[-2, 0]$ .

In generale si può notare che il numero di iterazioni nel metodo della secante è maggiore rispetto a quello del metodo di Newton, nonostante gli intervalli iniziali nel caso della secante siano stati scelti in modo da essere molto vicini alla radice. Ciò è dovuto al fatto che il metodo della secante converge con ordine del rapporto aureo, mentre il metodo di Newton converge con ordine quadratico per radici semplici.

(2d) Si è scelto di studiare l'errore della secante per la radice di  $f_1$ . Si riporta in figura 23 il grafico della successione  $d_k = |x_k - x_{k-1}|$ . Come stima di  $q$  si è scelto di considerare l'ultimo punto della successione, e si ottiene  $q = 1.692$ , sufficientemente in accordo con  $q_{atteso} = 1.618$ .

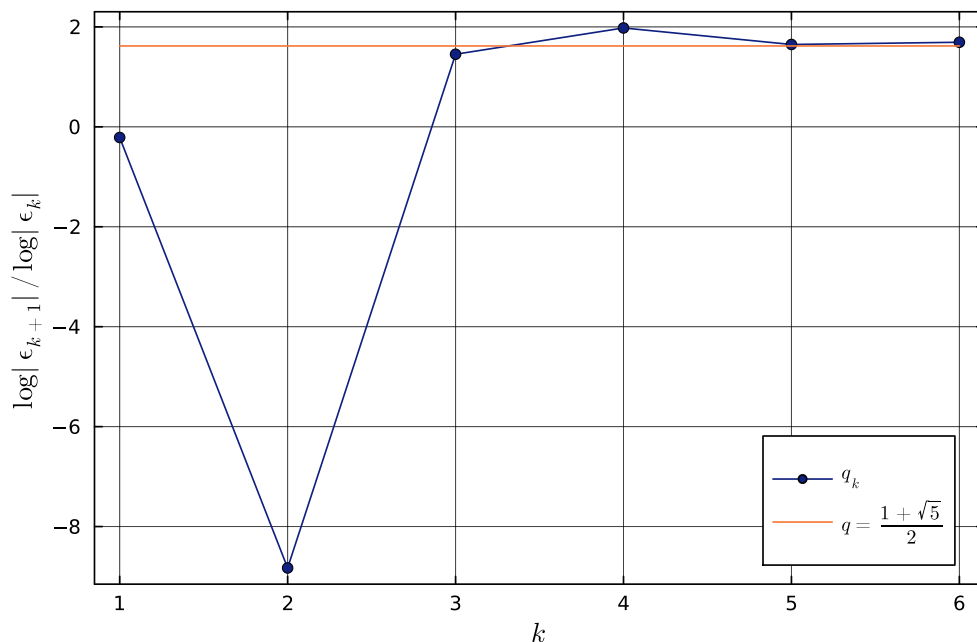


Figure 23: Studio del valore di  $q$  per la radice di  $f_1$ .

Radice	Radice stimata	Punti iniziali	Numero di iterazioni
$r_1$	1.068	[1.0, 2.0]	11
$r_2$	3.033	[2.0, 4.0]	8
$r_3$	6.308	[6.0, 7.0]	7
$r_4$	9.413	[9.0, 10.0]	7

Table 7: Radici stimate e punti iniziali per il metodo della secante.

### 3.8 Esercizio 3.4.3

Utilizza un grafico per localizzare approssimativamente tutte le radici di  $f(x) = x^{-2} - \sin(x)$  nell'intervallo  $[0.5, 10]$ . Poi trova una coppia di punti iniziali per ciascuna radice tale che il metodo della secante converga a quella radice.

#### 3.8.1 Soluzione

Si riporta in figura 24 il grafico della funzione  $f(x) = x^{-2} - \sin(x)$  nell'intervallo  $[0.5, 10]$ , da cui si evince che la funzione ha 4 radici nell'intervallo. In figura sono riportate anche le radici stimate.

Si riportano in tabella 7 le radici stimate, i punti iniziali scelti e il numero di iterazioni necessarie per la convergenza. Possiamo osservare che le radici trovate sono in accordo con quelle stimate nell'esercizio 3.4 con metodo di Newton.

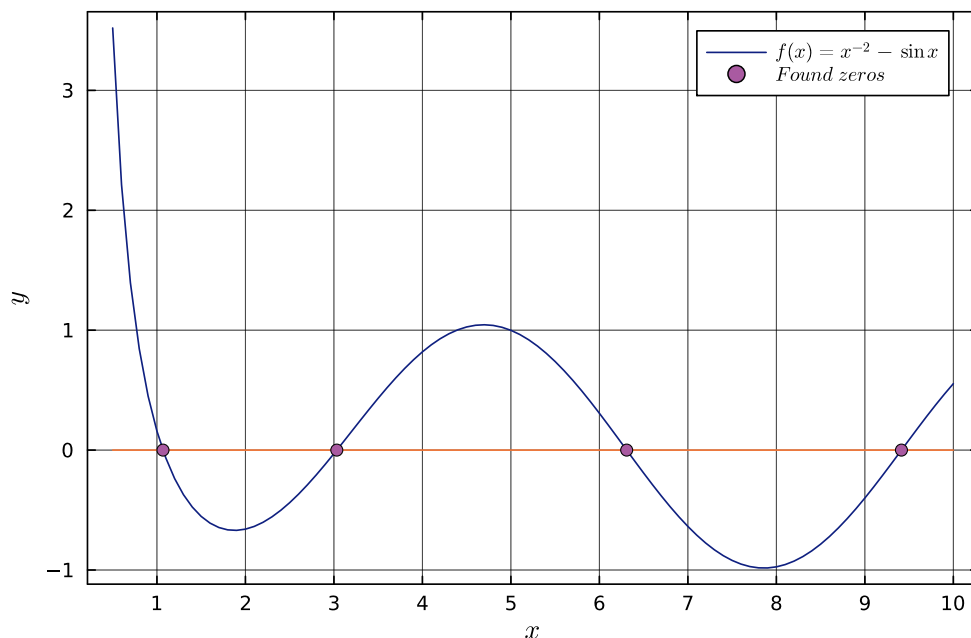


Figure 24: Grafico della funzione  $f(x) = x^{-2} - \sin(x)$  nell'intervallo  $[0.5, 10]$ .

## 4 Interpolazioni

### 4.1 Teoria

### 4.2 Esercizi 4.2.1, 4.2.2

1. Scrivi un programma che implementi la formula baricentrica dell'interpolazione di Lagrange per nodi in posizioni generiche.
2. In ciascun caso, interpola la funzione data usando  $n$  nodi equispaziati nell'intervallo indicato. Rappresenta graficamente ciascuna funzione interpolante insieme alla funzione esatta.
  - (a)  $f(x) = \ln(x)$ ,  $n = 2, 3, 4$ ,  $x \in [1, 10]$
  - (b)  $f(x) = \tanh(x)$ ,  $n = 2, 3, 4$ ,  $x \in [-3, 2]$
  - (c)  $f(x) = \cosh(x)$ ,  $n = 2, 3, 4$ ,  $x \in [-1, 3]$
  - (d)  $f(x) = |x|$ ,  $n = 3, 5, 7$ ,  $x \in [-2, 1]$

#### 4.2.1 Soluzione

(1) Si è implementata la formula baricentrica dell'interpolazione di Lagrange per nodi in posizioni generiche, e la si è testata nell'esercizio 4.2.2.

(2) In questo esercizio, nonostante sia data la possibilità di utilizzare la formula baricentrica semplificata dai nodi equispaziati, si è scelto di utilizzare la formula baricentrica per nodi generici, come test dell'implementazione dell'esercizio precedente. In esercizi successivi sarà richiesto di implementare la formula baricentrica per nodi di Chebyshev, e in tale con-

testo si è sviluppata una funzione Julia che chiede la natura dei nodi e restituisce la funzione baricentrica corrispondente, anche per il caso equispaziato. Si riportano i grafici delle funzioni interpolanti in figura 25.

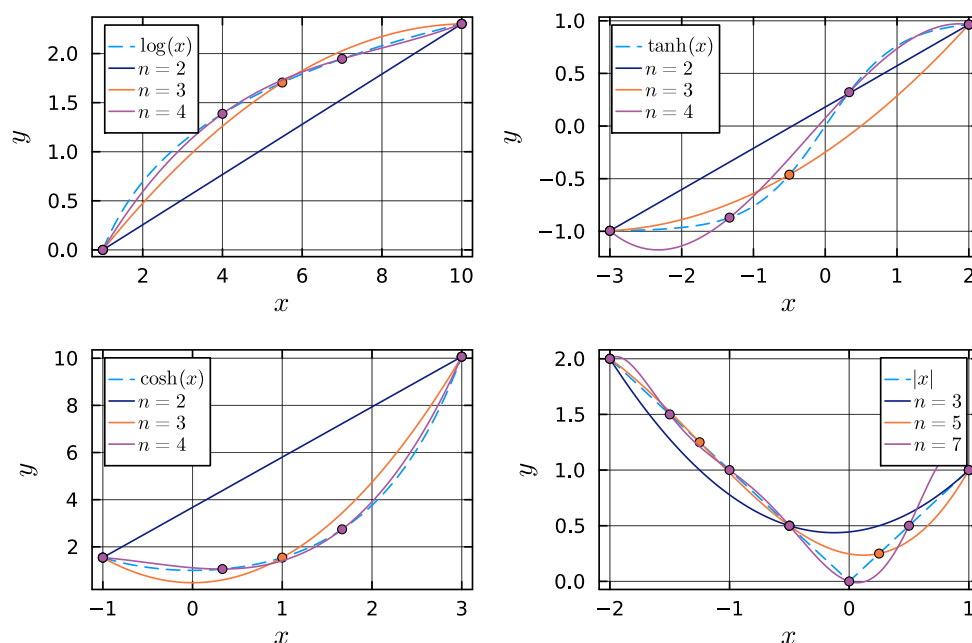


Figure 25: Grafici delle funzioni interpolanti per i vari casi.

Come si può notare, all'aumentare del numero di nodi l'interpolazione migliora. Il grafico della funzione  $|x|$  però richiede un numero di nodi maggiore per essere interpolato con una certa accuratezza, ciò è dovuto alla non derivabilità della funzione in  $x = 0$ . Il problema è che si sta cercando di interpolare una funzione non derivabile con un polinomio, che notoriamente non può farlo.

## 5 Integrazione numerica

### 5.1 Teoria

## 6 Equazioni differenziali ordinarie

### 6.1 Teoria

## 7 Appendice

### 7.1 Dati

Table 8: Distanza dal Sole e periodo orbitale dei pianeti del Sistema Solare

<b>Pianeta</b>	<b>Distanza dal Sole [Mkm]</b>	<b>Periodo orbitale [giorni]</b>
Mercurio	57.59	87.99
Venere	108.11	224.7
Terra	149.57	365.26
Marte	227.84	686.98
Giove	778.14	4332.4
Saturno	1427	10759
Urano	2870.3	30684
Nettuno	4499.9	60188