

Calcolo Numerico

Manuale per le ripetizioni di Informatica

Marco Lampis

24 gennaio 2023

Indice

0	Informazioni generali	1
0.1	Come svolgo le lezioni	1
1	Introduzione	3
2	Approssimazione della radice quadrata	5
2.1	Metodo Babilonese	5
3	Ricerca degli zeri di una funzione	7
3.1	Metodo della bisezione	7
4	PRNG	9

0 Informazioni generali

Ciao! Sono **Marco**, sono uno studente magistrale in *Software Engineering* al Politecnico di Torino e mi sono laureato in Ingegneria Informatica all'università di Pisa. Nella vita sono un programmatore, uno smanettone e amante di videogiochi! Amo quello che studio, e per questo motivo fornisco ripetizioni di informatica con particolare attenzione a:

- programmazione (Java, C++, C, Python, C#, Javascript, PHP)
- algoritmi e strutture dati
- basi di Dati
- più o meno tutto quello che riguarda l'informatica!

Sia per studenti delle scuole superiori che per l'università.

0.1 Come svolgo le lezioni

La mia metodologia è **innovativa** e **interattiva**: utilizzo una piattaforma apposita per lavorare contemporaneamente sullo stesso file con gli studenti, come se fossimo accanto. In questo modo, le lezioni sono più coinvolgenti e divertenti, e gli studenti possono imparare in modo facile ed efficiente. Inoltre, le lezioni includono sia esercitazioni che approfondimenti teorici, aiuto compiti e revisione.

Per aiutare gli studenti a prepararsi al meglio e a esercitarsi anche a casa, ho creato un sito web dedicato con risorse utili come teoria, esercizi e slide. Su questo sito, gli studenti possono trovare tutto il materiale di cui hanno bisogno, e il materiale è sempre disponibile gratuitamente per i miei studenti. Inoltre, uso un iPad per prendere appunti durante le lezioni e poi condividerli con gli studenti, in modo che possano rivedere tutto ciò che è stato trattato.

Se volete migliorare le vostre conoscenze in informatica e avere un supporto personalizzato e professionale, non esitate a contattarmi. Sarò felice di fornirvi maggiori informazioni e di fissare una lezione con voi.

1 Introduzione

Il calcolo numerico è una branca dell'informatica che si occupa di risolvere problemi matematici mediante l'utilizzo di algoritmi e tecniche computazionali. In pratica, il calcolo numerico si occupa di trovare soluzioni approssimate di problemi matematici che **non possono essere risolti in modo esatto** mediante metodi analitici tradizionali.

Il calcolo numerico utilizza una vasta gamma di tecniche e algoritmi, tra cui:

- Metodi iterativi per la risoluzione di sistemi di equazioni non lineari
- Metodi di interpolazione e approssimazione per la costruzione di modelli matematici
- Metodi di integrazione e di risoluzione di equazioni differenziali
- Algoritmi di ottimizzazione per la ricerca di soluzioni ottimali.

Il calcolo numerico è ampiamente utilizzato in molte applicazioni, tra cui l'informatica, la fisica, l'ingegneria e molto altro.

2 Approssimazione della radice quadrata

Per calcolare la radice quadrata di un numero, si utilizzano diverse tecniche di approssimazione, come la scelta di un valore approssimato per la radice.

2.1 Metodo Babilonese

Il **metodo babilonese** per il calcolo della radice quadrata di un numero N consiste in una serie di approssimazioni successive. Inizialmente si sceglie un numero casuale (detto approssimazione iniziale) che sia maggiore o uguale a N . Successivamente si utilizza la seguente formula per ottenere una nuova approssimazione:

$$x = (x + N/x)/2$$

dove x è l'approssimazione corrente.

Il processo viene ripetuto utilizzando la nuova approssimazione come input fino ad ottenere una precisione desiderata denominata ϵ .

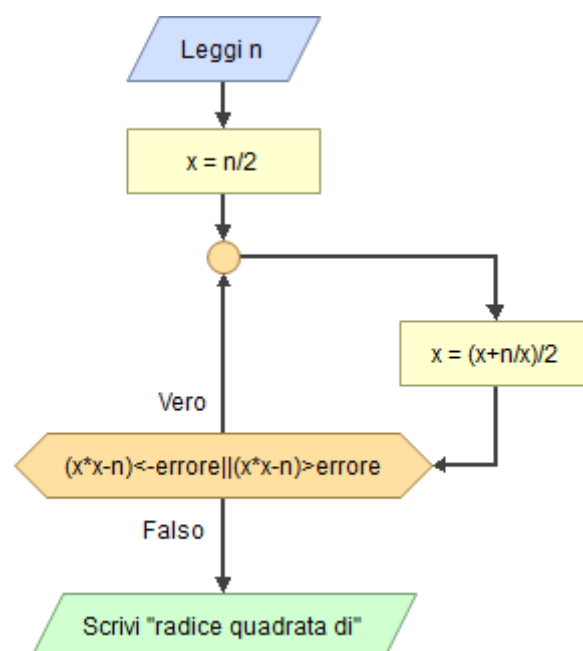
In pratica si ripete la formula finché la differenza tra l'approssimazione corrente e quella precedente sia minore di una certa soglia, come mostrato:

$$|x_{n+1} - x_n| < \epsilon$$

Un ulteriore criterio potrebbe essere:

$$\frac{|x_{n+1} - x_n|}{x_n} < \epsilon$$

E' un metodo molto efficace per calcolare la radice quadrata di un numero, soprattutto perché si basa su una serie di approssimazioni che tendono ad avvicinarsi sempre di più alla soluzione vera.

**Figura 2.1:** Esempio algoritmo

3 Ricerca degli zeri di una funzione

Trovare gli zeri di una funzione è un calcolo anch'esso complicato che richiede l'approssimazione delle soluzioni per trovarne l'esito.

3.1 Metodo della bisezione

Il **metodo di bisezione** è un algoritmo utilizzato per trovare gli zeri di una funzione in un determinato intervallo. L'idea di base del metodo è quella di dividere continuamente l'intervallo di ricerca in due parti uguali e di determinare in quale metà si trova lo zero.

Per utilizzare tale metodo è necessario che: - la funzione sia **continua**. - gli estremi siano **discontinui**, altrimenti non sarebbe possibile dire con certezza che esiste uno zero (*teorema degli zeri*).

Per utilizzare il metodo di bisezione, è necessario:

- specificare un intervallo iniziale (ad esempio $[a, b]$) in cui si ritiene che si trovi lo zero della funzione.
- Successivamente, si calcola il punto medio del intervallo ($c = (a + b)/2$).
- A questo punto, si valuta il segno della funzione nel punto medio (c) e nel punto iniziale (a). Se i segni sono diversi, significa che lo zero della funzione si trova nell'intervallo $[a, c]$, altrimenti si trova in $[c, b]$.

Il processo si può ripetere fino a quando il valore di c non sia sufficientemente vicino allo zero desiderato, o una sua approssimazione.

Si può affermare che il metodo converge allo zero in quanto la formula usata per il calcolo del punto medio $\frac{(b-a)}{2^n}$ con n tentativi che tende a $+\infty$ fa convergere la frazione a 0.

3.1.1 Algoritmo

Una versione estesa dell'algoritmo in pseudocodice è la seguente:

```
1 var cx;  
2  
3 do {  
4     var cx = (sx + dx) / 2;  
5  
6     if( f(sx) * f(cx) < 0 )  
7         dx = cx;  
8  
9     else {  
10        if (f(cx)*f(dx) < 0)  
11            sx = cx;  
12  
13        else  
14            return cx;  
15    }  
16  
17 } while(dx - sx > precisione);
```

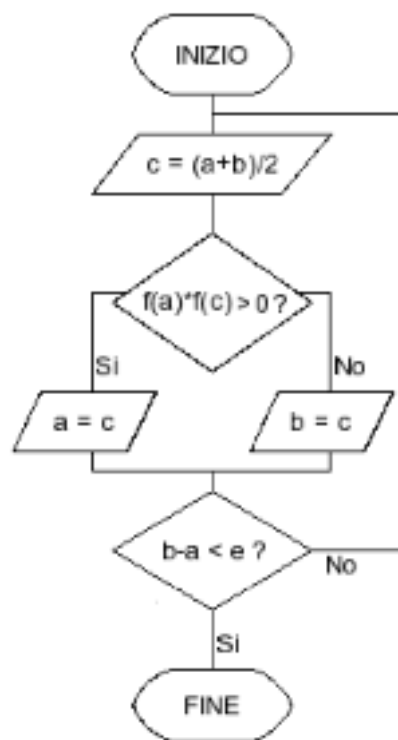


Figura 3.1: Algoritmo bisezione

4 PRNG

Il generatore di numeri casuali pseudo-casuali (PRNG) è un algoritmo che utilizza una seme iniziale per generare una sequenza di numeri che appaiono casuali, ma in realtà sono prevedibili se si conosce il seme. Il generatore utilizza un algoritmo matematico per generare una successione di numeri che sembrano essere estratti in modo casuale, emulandone le proprietà statistiche.

I PRNG sono utilizzati in molte applicazioni, tra cui la crittografia, i giochi di simulazione, la generazione di numeri casuali per i test statistici e molto altro ancora.

