LABORATORIO DI CALCOLO – CALCOLO DI INTEGRALI DEFINITI

Marco Vecchio

20044258

1. **Introduzione**

Il progetto richiede la realizzazione di un programma in C che calcoli l’integrale definito di una funzione assegnata tramite due metodi tra quelli visti a lezione.

I metodi scelti sono stati: ***il metodo composito del trapezio*** e ***il metodo composito di Cavlieri/Simpson***.

La funzione assegnata è invece: *sqrt(sin(x))*  considerata nell’intervallo [0,π].

1. **I metodi**

* **Metodo del trapezio**

Questo metodo approssima l’area sottesa dalla funzione con l’area di un trapezio.

Data una funzione f e un intervallo [a,b] l’integrale è calcolato secondo la formula:

Questo metodo ha un grado di precisione abbastanza basso, infatti risulta evidente che utilizzare un solo trapezio generi un errore piuttosto grande. Per questo motivo si migliora l’accuratezza dividendo l’intervallo in *n* parti aventi tutte la stessa ampiezza *h*, la quale è calcolta come:

Si ottengono quindi i unti di suddivisione:

Per tutti questi punti è possibile calcolare il valore della funzione

Applicando su tutti questi intervalli la formula vista in precedenza è possibile calcolare in maniera meglio appprossimata l’area:

* **Metodo di Cavalieri/Simpson**

Questo metodo permette di calcolare l’area sottesa dalla funzione approssimandola con l’area sottesa da un arco di parabola.

Data una funzione f e un intervallo [a,b] l’integrale viene clacolato nel seguente modo:

Come per il metodo precedente si può dividere allo stesso modo l’intevallo in sottointervalli, si può anche calcolare il punto medio di ogni intervallo j come:

Detto ciò si può ricalcolare l’integrale come:

Ricordandosi di ‘a’ e ‘b’ la formula diventa:

Si può poi riorganizzare i calcoli introducendo:

Da cui si ottiene la formul finale:

1. **Realizzazione progetto**

* **Metodo composito del trapezio**

Per l’implementazione mi sono servito di un calcolo iterativo, prima avviene la suddivisione in parti nT = 1 poi:

* Viene calcolato l’integrale con nT suddivisioni
* Viene calcolato l’integrale con 2\*nT suddivisioni
* Viene calcolata la differenza tra le due approsimazioni, se questa è inferiore alla tolleranza il calcolo si ferma, altrimenti il numero delle suddivisioni nT viene raddoppiato

Questa è la porzione di codice che realizza quanto appena detto:

Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente

In cui viene chiamata due volte la seguente funzione trapezio che si occupa di calcolare l’integrale:

Immagine che contiene testo, schermata, software, schermo

Descrizione generata automaticamente

Dove la funzione riceve in input: la funzione data, i due estremi dell’intervallo e il numero di suddivisioni.

* **Metodo composito di Cavalieri/Simpson**

L’implementazione di questo metodo ha la stessa struttura del precedente, anche qui c’è un ciclo while che ha le stesse condizioni di quello precedente e termina nello stesso modo.

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

L’unica diffrenza è che in questo caso la suddivisione è nS=2.

Questa è invece l’implementazione della funzione simpson per il calcolo:

Immagine che contiene testo, schermata, software

Descrizione generata automaticamente

La quale riceve in input le stesse cose del metodo precendente.

1. **Grafici**

La funzione che mi è stata assegnata è *sqrt(sin(x))*, qui presento il suo grafico realizzato tramite **GNUPlot**:

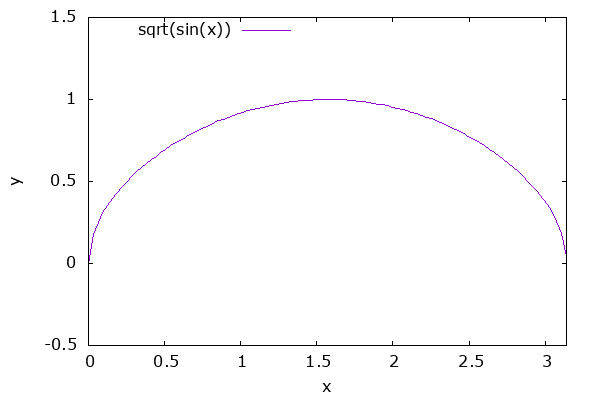


Figura 1 - funzione sqrt(sin(x))

Ho inoltre analizzato i dati provenienti dal programma e li ho rappresnetati mediante un grafico. Per il calcol sono state usate accuratezze sempre più precise: 0.5,0.05,0.005,0.0005 e 0.00005.

Come si può notare il metodo del trapezio è inizialmente leggermente più preciso, ma i due metodi diventano accurati in maniera molto similare.

Immagine che contiene testo, linea, diagramma, Diagramma

Descrizione generata automaticamente

Figura 2 - Integrale della funzione con i due metodi