Relazione di Laboratorio di Calcolo

Riccardo Nazzari 20044680, Linda Monfermoso 20028464

Primo Semestre A.A. 2023/2024

1 Introduzione al progetto

Il progetto assegnato prevede la scrittura di un programma in linguaggio C per il calcolo di integrali definiti, utilizzando due metodi diversi e fornendo i risultati con almeno 4 cifre decimali esatte. Abbiamo deciso di determinare il valore delle funzioni integrate con il **metodo composito del punto medio** ed il **metodo composito di Cavalieri/Simpson**.

2 Richiami di teoria

2.1 Integrazione numerica

L'integrale è un operatore lineare che, data una funzione f(x) e un intervallo [a, b] nel dominio, restituisce il valore dell'area sottesa nel suo grafico.

$$I(f) = \int_a^b f(x)\,dx = F(b) - F(a)$$

Questo calcolo risulta facile nel momento in cui siamo a conoscenza della primitiva F, ma non sempre è così: quando la funzione f(x) viene definita come integrale (come la funzione di Eulero) oppure è nota solo in alcuni punti bisogna ricorrere alla **integrazione numerica** dell'integrale.

L'integrazione numerica, detta anche *quadratura numerica*, consiste nello stimare il valore di un integrale definito, con il vantaggio di poterlo fare senza conoscere la funzione primitiva. Del resto sappiamo che l'integrale non è altro che il calcolo dell'area della superfcie delimitata dal grafico della funzione, dall'asse delle ascisse e dalle rete di equazione x=a e x=b; di fatto, i metodi esistenti approssimano il calcolo di tale area mediante diverse tecniche.

I metodi per l'integrazione numerica si dividono in due macrocategorie:

- **Formule di Newton-Cotes**: si basano sulla valutazione dell'integrando in n+1 punti equidistanti e sono consigliate da utilizzare se il valore dell'integrando nei punti è noto. Fanno parte di questa famiglia il **metodo del punto medio**, il **metodo di Cavalieri/Simpson** e la **formula del trapezio**;

- **Formule di Gauss**: sono preferibili da utilizzare quando è possibile modificare i punti dove è valutato l'integrando, basandosi sulla conoscenza di n+1 valori della funzione nell'intervallo considerato.

Nel nostro progetto abbiamo impiegato il metodo del punto medio e di Cavalieri/Simpson in versione composita.

2.2 Metodo del punto medio

Conosciuto anche come **metodo dei rettangoli**, è il modo più semplice per approssimare un integrale. Questo metodo possiede un grado di precisione molto basso, ed il calcolo si basa sul rappresentare l'integrale come un rettangolo, formato da:

- una base di valore (b-a), dove 'b' e 'a' sono i due estremi di integrazione;
- un'altezza di valore f(c), dove 'c' è il punto medio dell'intervallo.

Otteniamo quindi un'espressione dell'integrale pari a:

$$I_r = (b-a)\, f(c) \;=\; (b-a) f\left(\frac{a+b}{2}\right)$$

2.2.1 Formula composita del punto medio

L'errore che si ottiene con il metodo del punto medio appena enunciato è abbastanza alto: in particolare, è evidente come l'utilizzo di *un solo* rettangolo lasci spazio ad un errore di precisione abbastanza grossolano rispetto al risultato sperato. È possibile aumentare l'accuratezza di questo metodo dividendo l'intervallo in n parti con la stessa ampiezza h, calcolata come:

$$h = \frac{b-a}{n}$$

da cui ricaviamo i punti di suddivisione:

$$x_0 = a,\ x_1 = a + h,\ x_2 = a + 2h,\ \ldots x_n = a_n + nh = b.$$

su cui possiamo calcolare i valori della funzione:

$$y_0 = f(a),\ y_1 = f(x_1),\ y_2 = f(x_2),\ \ldots, y_{n-1} = f(x_{n-1}),\ y_n = f(b).$$

Quelli che otteniamo sono dei rettangoli che hanno come base l'intervallo di suddivisione, mentre come altezza il segmento rappresentato dal valore di f calcolato nel primo estremo oppure nel secondo.

A questo punto possiamo calcolare l'area applicando la formula vista in precedenza per ogni intervallo, ovvero:

$$\frac{(b-a)}{n} \sum_{i=0}^{n-1} f(x_i).$$

2.3 Metodo di Cavalieri/Simpson

Conosciuto anche come **metodo delle parabole** consiste nell'approssimare il grafico della funzione con un arco di parabola. Data una funzione f e un intervallo [a, b], possiamo scrivere:

$$\int_a^b f(x)dx \approx \frac{(b-a)}{6}\left(f(a) + 4f\left(\frac{(a+b)}{2}\right) + f(b)\right)$$

2.3.1 Metodo composito di Cavalieri/Simpson

Facendo lo stesso ragionamento come per il metodo dei rettangoli, possiamo pensare di suddividere l'intervallo [a, b] in n parti: definiamo m_j il punto medio del j-esimo intervallo come:

$$m_j = a + \frac{2j-1}{2}h$$

possiamo riscrivere l'integrale di partenza come:

$$\int_a^b f(x)dx = \sum_{j=1}^{n} \frac{n}{6}[f(x_{j-1}) + 4f(m_j) + f(x_j)]$$

di cui, ricordandoci dei punti 'a' e 'b', possiamo semplificare come:

$$\frac{h}{6}\left[f(a) + 4\sum_{j=1}^{n} f(m_j) + 2\sum_{j=1}^{n-1} f(x_j) + f(b)\right]$$

A questo punto, possiamo introdurre $z_k = a + k\frac{b-a}{2n}$ per cui raggiungiamo la formula finale:

$$\int_a^b f(x)dx \approx \frac{(b-a)}{6n}[f(z_0) + 4f(z_1) + 2f(z_2) + 4f(z_3) + 2f(z_4) + ... + f(z_{2n})]$$

Questa nuova formula ci consente una maggiore precisione nel calcolo.

3 Sviluppo del progetto

3.1 Metodo composito del punto medio

Per implementare il metodo composito del punto medio ci siamo serviti di un semplice calcolo iterativo in cui, dopo aver stabilito una suddivisione in parti di n=1:

1. viene calcolato l'integrale approssimato con n suddivisioni;
2. viene calcolato l'integrale approssimato con $2n$ suddivisioni;
3. viene calcolata la differenza tra i due valori: se questa è inferiore alla tolleranza prefissata ci si ferma, altrimenti il numero delle suddivisioni viene raddoppiato.

Il valore dell'integrale viene calcolato con il seguente frammento di codice:

```
float puntoMedio(float funzione(float), float a, float b, int n) {
 float somma = 0.;
 float h, x_i, x_i1;

 h = (b-a)/n;
 for (int i = 0; i < n; i++) {
     x_i = a+i*h;
     x_i1 = x_i+h;
     somma = somma + funzione((x_i+x_i1)/2);
 }
 somma = somma*h;
 return somma;
}
```

dove in input abbiamo: la funzione da valutare, i due punti estremi e il numero di suddivisioni. Questo metodo viene richiamato all'interno di un ciclo while ogni volta che la condizione del punto 3 viene rispettata.

```
n = 1;
while(1) {
   switch(s) {
       case 1:
           in = puntoMedio(coseno, a, b, n);
           i2n = puntoMedio(coseno, a, b, 2*n);
           break;

       [...]

   }
   if((fabs(in-i2n) < acc))
       break;
   n = 2*n;
}
```

3.2 Metodo composito di Cavalieri/Simpson

La strategia per implementare questo metodo si basa esattamente su quella appena esposta per il punto medio, ovvero sfruttando un semplice calcolo iterativo: l'unica differenza si trova nella suddivisione in parti che, per Cavalieri/Simpson, è m=2.

Il valore dell'integrale viene calcolato con il seguente frammento di codice:

```
float cavalieriSimpson(float funzione(float), float a, float b, int
    m) {
 float somma = 0.;
 float h, x_i;

 h = (b-a)/m;
 for (int i = 0; i <= m; i++) {
     x_i = a+i*h;
     if (i == 0 || i == m)
         somma = somma + funzione(x_i);
     else {
         if (i%2 == 0)
             somma = somma + 2*funzione(x_i);
         else
             somma = somma + 4*funzione(x_i);
     }
 }

 somma = somma*h/3;
 return somma;
}
```

i cui input sono gli stessi medesimi del metodo del punto medio. Anche qui abbiamo un ciclo while, che termina nel momento in cui la differenza tra i valori dei due integrali calcolati è minore dell'accuratezza.

```
m = 2;
 while(1) {
     switch(s) {
         case 1:
             im  = cavalieriSimpson(coseno, a, b, m);
             i2m = cavalieriSimpson(coseno, a, b, 2*m);
             break;

         [...]

     }
     if((fabs(im-i2m) < acc))
         break;
     m = 2*m;
 }
```

3.3 Error checking

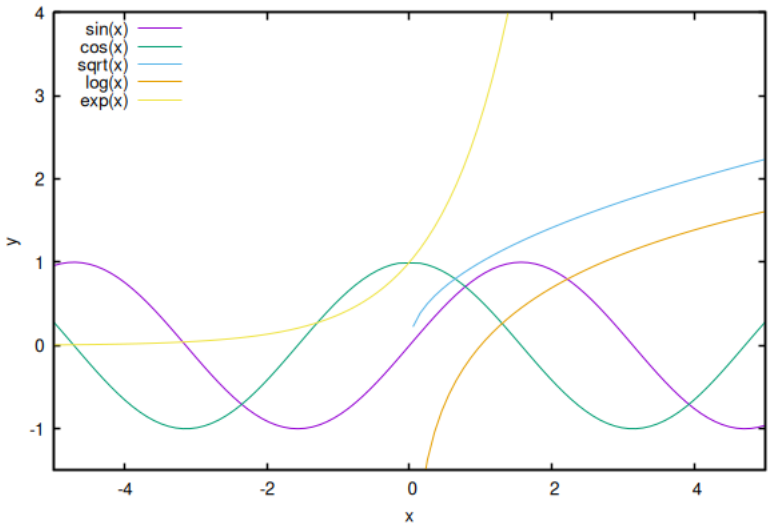
Per ogni richiesta di input è stata implementata una fase di **error checking**, in modo che i caratteri non consentiti non siano presi dal programma. Per esempio:

```
printf("Inserire il primo estremo:\n");
printf("> ");
do {
   scanf("%s", A);
   a = strtof(A, NULL);
   if(strcmp(A, "0") == 0)
         break;
   if(a == 0.0F) {
      printf("Inserire un numero.\n");
      printf("> ");
      }
   } while(a == 0.0F);
```

4 Grafici e osservazioni

4.1 Funzioni analizzate

Abbiamo preso in esame le funzioni **coseno**, **seno**, **radice quadrata**, **logaritmo** e **esponenziale**, fornite dalla libreria `math.h` del C. Presentiamo qui i loro grafici, realizzati con l'ausilio del software GNUPlot:



4.2 Risultati dell'integrazione numerica

Abbiamo inoltre analizzato i dati pervenuti dal programma e li abbiamo illustrati tramite grafici. Le precisioni utilizzate sono: 0.5, 0.05, 0.005, 0.0005, 0.00005.

Per quanto riguarda le funzioni seno, coseno e logaritmo, possiamo osservare come l'integrazione tramite metodo del punto medio divenga accurata solo quando viene utilizzata una precisione molto alta (ossia un numero prossimo allo zero), mentre con il metodo di Cavalieri/Simpson risulti più accurata anche con precisione relativamente bassa.

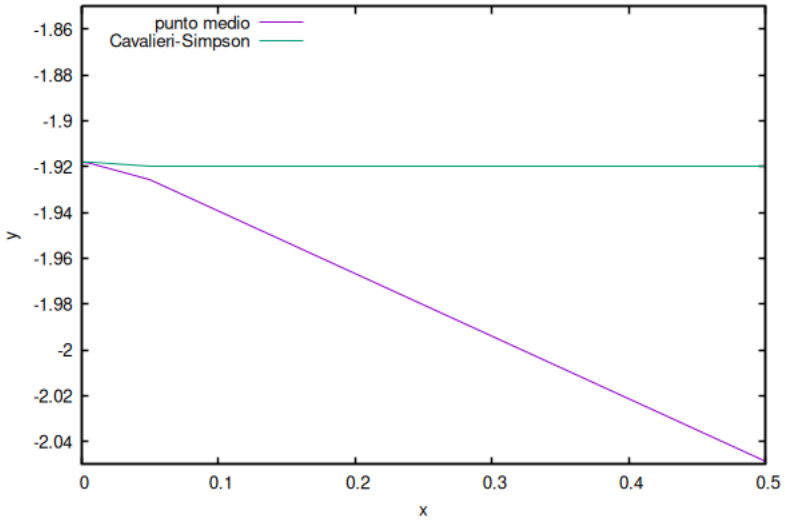


Figure 1: Integrazione della funzione $\cos x$ (range: -5, 5)

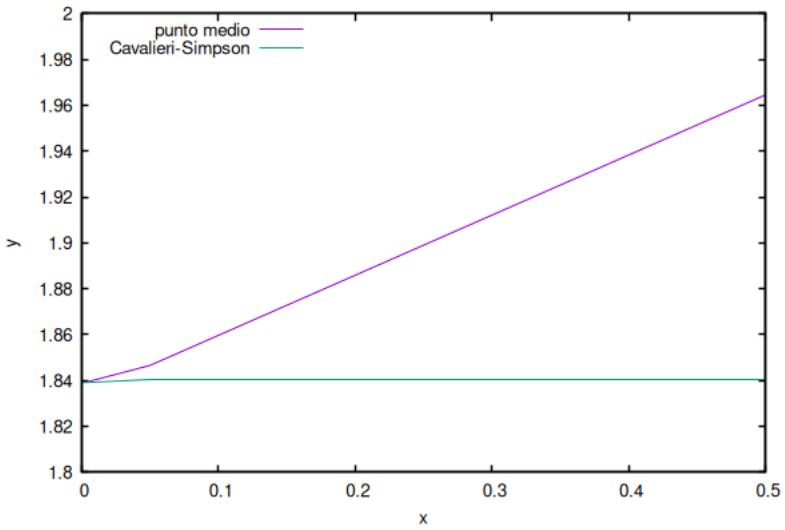


Figure 2: Integrazione della funzione $\sin x$ (range: 0, 10)

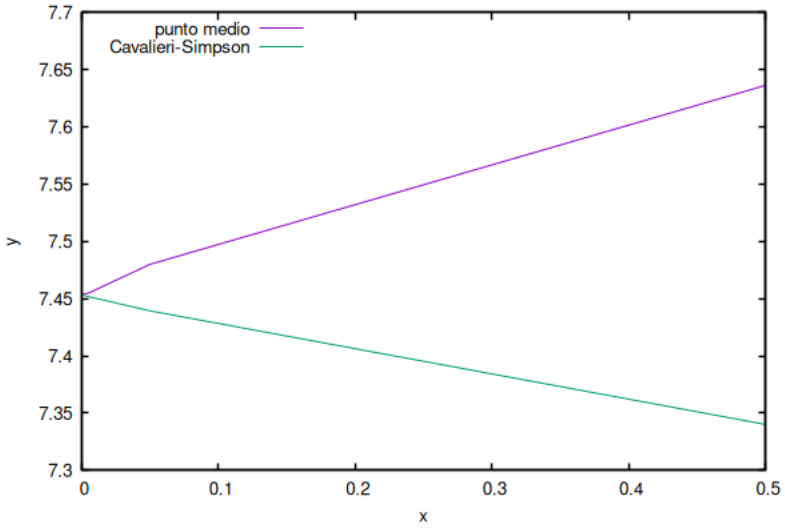


Figure 3: Integrazione della funzione $\sqrt{x}$ (range: 0, 5)

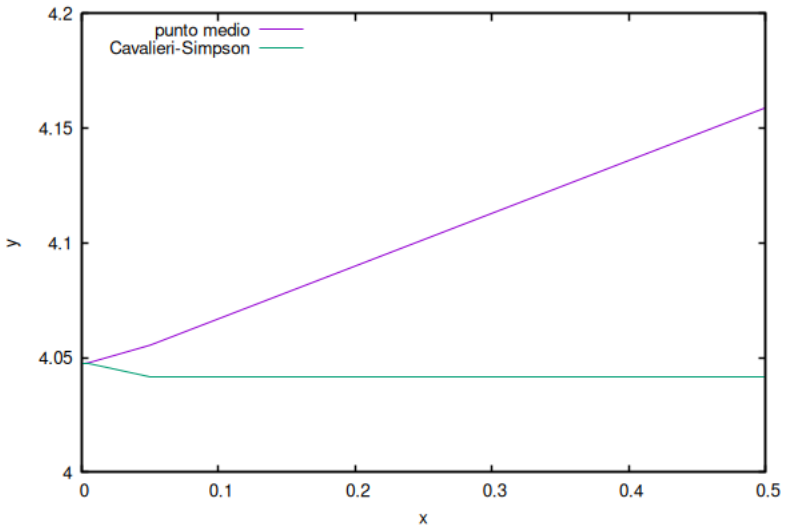


Figure 4: Integrazione della funzione $\log x$ (range: 1, 5)

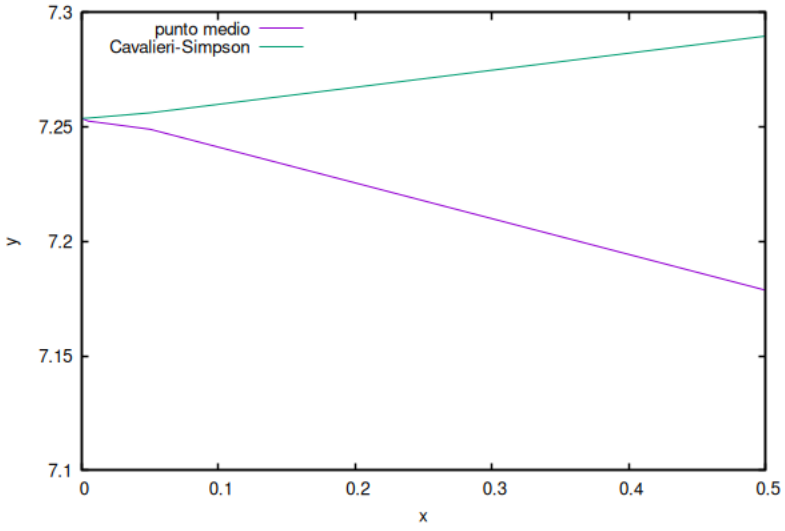


Figure 5: Integrazione della funzione $\exp x$ (range: -2, 2)