Laboratorio ALAN Errori

Biberia Mario Madalin 5608210
Muceku Denis 4801139
Vassallo Eugenio 5577783

Relazione sul Laboratorio sugli errori in C++

1. **Confronto analitico delle operazioni equivalenti implementate attraverso diversi algoritmi**
2. **implementato il polinomio di Taylor per la funzione e^x e confronto con esso**
3. **Realizzazione di un programma che valuta la precisione della macchina in precisione singola e doppia.**

Per Consultare il codice c++ vedere i file “1/2/3esercizio”.cpp
Per Consultare i risultati relativi a ogni punto vedere il file “erroriout.txt” prodotto dopo l'esecuzione del codice usando make il txt contiene i risultati di tutti i punti

Esercizio 1)

Durante il laboratorio sono stati confrontati due algoritmi equivalenti per il calcolo della somma di numeri:

- $(a+b)+c$
- $a+(b+c)$

Il primo algoritmo ha evidenziato il fenomeno della cancellazione numerica

il secondo è risultato non ha questo problema, in questo caso.

Motivazione: quando si sommano due numeri in virgola mobile di grandezze molto diverse, il risultato potrebbe non mostrare alcuna variazione significativa a causa della differenza di scala. Questo effetto si verifica perché, nell'aritmetica dei computer, i numeri sono rappresentati con precisione limitata, quindi non tutti i numeri reali possono essere rappresentati esattamente. Di conseguenza, si possono verificare errori di arrotondamento.

Alla luce di queste informazioni abbiamo capito che la scelta dell'algoritmo influisce significativamente sui risultati, poiché la condizione del problema e la stabilità numerica dell'algoritmo determinano la precisione delle soluzioni.

Un problema è considerato ben condizionato se piccole variazioni nei dati di ingresso producono solo lievi variazioni nei risultati; altrimenti, è considerato mal condizionato.

Esercizio 2)

Abbiamo implementato il polinomio di Taylor per la funzione e^x

$$f_N(x) = \sum_{n=0}^{N} \frac{x^n}{n!}$$

Abbiamo osservato che l'accuratezza dell'approssimazione aumenta con il crescere del termine N.

Invece per x molto grandi, la precisione del polinomio di Taylor diminuisce,in particolare per valori negativi di x.

Algoritmo 1:

Per valori di x vicini a zero, gli errori sono ridotti, e migliorano ulteriormente all'aumentare di n. Quando x si allontana da zero, gli errori aumentano, ma tendono comunque a diminuire al crescere di n. Tuttavia, si osserva che gli errori per x=-30 sono significativamente elevati. Questo accade perché la funzione di Taylor approssima l'esponenziale tramite una somma di potenze con esponenti alternativamente pari e dispari, il che porta a risultati falsati quando x è negativo e lontano da zero.

Algoritmo 2:

Per valori di x vicini a zero, gli errori calcolati dall'algoritmo 2 sono simili a quelli dell'algoritmo 1. il miglioramento si manifesta più chiaramente quando ci si allontana da zero. Ad esempio, per x=−30, i risultati prodotti dall'algoritmo 2 hanno meno errori rispetto a quelli dell'algoritmo 1

Entrambi gli algoritmi migliorano con l'aumentare di N, ma l'algoritmo 2 offre vantaggi evidenti per valori di x negativi.

Esercizio 3)

La precisione di macchina è stata determinata empiricamente usando un ciclo `while` che continua a dividere 1 finché la differenza tra il valore corrente e il precedente diventa inferiore alla precisione di macchina.

Questo comportamento indica che siamo oltre il limite della precisione rappresentabile epsilon. Il numero di cicli necessari è stato registrato, permettendo di determinare il valore di d.

[float è a singola precisione è double a doppia precisione]

Risultati ottenuti:

- Machine epsilon for single precision: 1.19209e-07 = 2^-d where d is: 23
- Machine epsilon for double precision: 2.22045e-16 = 2^-d where d is: 52

Esercizio 1:

For i = 0, (a + b) + c = 0, a + (b + c) = 1
For i = 1, (a + b) + c = 0, a + (b + c) = 10
For i = 2, (a + b) + c = 0, a + (b + c) = 100
For i = 3, (a + b) + c = 0, a + (b + c) = 1000
For i = 4, (a + b) + c = 0, a + (b + c) = 10000
For i = 5, (a + b) + c = 98304, a + (b + c) = 100000
For i = 6, (a + b) + c = 1.01581e+06, a + (b + c) = 1e+06

Esercizio 2:

algoritmo 1

Valore x = 0.5
f(x) = 1.64872

fN(x) = 1.625 con N = 3
Errore assoluto 1.625 - 1.648721271 = -0.0237212707
Errore relativo -0.01438767797

fN(x) = 1.64872127 con N = 10
Errore assoluto 1.64872127 - 1.648721271 = -2.818771883e-10
Errore relativo -1.70967157e-10

fN(x) = 1.648721271 con N = 50
Errore assoluto 1.648721271 - 1.648721271 = -4.440892099e-16
Errore relativo -2.693537214e-16

fN(x) = 1.648721271 con N = 100
Errore assoluto 1.648721271 - 1.648721271 = -4.440892099e-16
Errore relativo -2.693537214e-16

fN(x) = 1.648721271 con N = 150
Errore assoluto 1.648721271 - 1.648721271 = -4.440892099e-16
Errore relativo -2.693537214e-16

Valore x = 30
f(x) = 1.068647458e+13

fN(x) = 481 con N = 3
Errore assoluto 481 - 1.068647458e+13 = -1.068647458e+13
Errore relativo -1

fN(x) = 76106409.57 con N = 10
Errore assoluto 76106409.57 - 1.068647458e+13 = -1.068639848e+13
Errore relativo -0.9999928782

fN(x) = 1.068092946e+13 con N = 50
Errore assoluto 1.068092946e+13 - 1.068647458e+13 = -5545120425
Errore relativo -0.0005188914625

fN(x) = 1.068647458e+13 con N = 100
Errore assoluto 1.068647458e+13 - 1.068647458e+13 = 0.00390625
Errore relativo 3.655321472e-16

fN(x) = 1.068647458e+13 con N = 150
Errore assoluto 1.068647458e+13 - 1.068647458e+13 = 0.00390625
Errore relativo 3.655321472e-16

Valore x = -0.5
f(x) = 0.6065306597

fN(x) = 0.625 con N = 3
Errore assoluto 0.625 - 0.6065306597 = 0.01846934029
Errore relativo 0.03045079419

fN(x) = 0.6065306595 con N = 10
Errore assoluto 0.6065306595 - 0.6065306597 = -2.573727897e-10
Errore relativo -4.24335993e-10

fN(x) = 0.6065306597 con N = 50
Errore assoluto 0.6065306597 - 0.6065306597 = -1.110223025e-16
Errore relativo -1.830448316e-16

fN(x) = 0.6065306597 con N = 100
Errore assoluto 0.6065306597 - 0.6065306597 = -1.110223025e-16
Errore relativo -1.830448316e-16

fN(x) = 0.6065306597 con N = 150
Errore assoluto 0.6065306597 - 0.6065306597 = -1.110223025e-16
Errore relativo -1.830448316e-16

Valore x = -30
f(x) = 9.357622969e-14

fN(x) = 421 con N = 3
Errore assoluto 421 - 9.357622969e-14 = 421
Errore relativo 4.499005799e+15

fN(x) = -41468364.71 con N = 10
Errore assoluto -41468364.71 - 9.357622969e-14 = -41468364.71
Errore relativo -4.431506255e+20

fN(x) = -1482183041 con N = 50
Errore assoluto -1482183041 - 9.357622969e-14 = -1482183041
Errore relativo -1.583931139e+22

fN(x) = -0.0002425284333 con N = 100
Errore assoluto -0.0002425284333 - 9.357622969e-14 = -0.0002425284334
Errore relativo -2591773939

fN(x) = -0.0002425283907 con N = 150
Errore assoluto -0.0002425283907 - 9.357622969e-14 = -0.0002425283908
Errore relativo -2591773484

algoritmo 2

Valore x = -0.5
f(x) = 0.6065306597

1/fN(x) = 0.6153846154 con N = 3
Errore assoluto 0.6153846154 - 0.6065306597 = 0.008853955672
Errore relativo 0.01459770505

1/fN(x) = 0.6065306598 con N = 10
Errore assoluto 0.6065306598 - 0.6065306597 = 1.036968289e-10
Errore relativo 1.709671675e-10

1/fN(x) = 0.6065306597 con N = 50
Errore assoluto 0.6065306597 - 0.6065306597 = 1.110223025e-16
Errore relativo 1.830448316e-16

1/fN(x) = 0.6065306597 con N = 100
Errore assoluto 0.6065306597 - 0.6065306597 = 1.110223025e-16
Errore relativo 1.830448316e-16

1/fN(x) = 0.6065306597 con N = 150
Errore assoluto 0.6065306597 - 0.6065306597 = 1.110223025e-16
Errore relativo 1.830448316e-16

Valore x = -30
f(x) = 9.357622969e-14

1/fN(x) = 0.002079002079 con N = 3
Errore assoluto 0.002079002079 - 9.357622969e-14 = 0.002079002079
Errore relativo 2.221720287e+10

1/fN(x) = 1.313949779e-08 con N = 10
Errore assoluto 1.313949779e-08 - 9.357622969e-14 = 1.313940421e-08
Errore relativo 140413.9091

1/fN(x) = 9.36248108e-14 con N = 50
Errore assoluto 9.36248108e-14 - 9.357622969e-14 = 4.858111501e-17
Errore relativo 0.0005191608507

1/fN(x) = 9.357622969e-14 con N = 100
Errore assoluto 9.357622969e-14 - 9.357622969e-14 = -3.786532345e-29
Errore relativo -4.046468166e-16

1/fN(x) = 9.357622969e-14 con N = 150
Errore assoluto 9.357622969e-14 - 9.357622969e-14 = -3.786532345e-29
Errore relativo -4.046468166e-16

Esercizio 3:

Machine epsilon for single precision: 1.19209e-07 = 2^-d where d is: 23
Machine epsilon for double precision: 2.22045e-16 = 2^-d where d is: 52