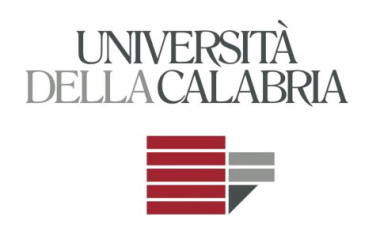
**RELAZIONE PROGETTO INTEGRAZIONE NUMERICA-**

**METODO SIMPSON/TRAPEZI**

****

**STUDENTE**: **De Fazio Francesco**

**DOCENTE**: W. Spataro

**UNIVERSITA’**: Università della Calabria

**DIPARTIMENTO**: Matematica e Informatica

**CORSO**: Algoritmi Paralleli e Sistemi Distribuiti

**A.A**: 2020/2021

Metodo dei trapezi

la regola del trapezio approssima l'integrale ( se la funzione è non negativa, l'area della regione piana compresa fra il grafico della funzione  e l'asse delle ascisse) con l'area del trapezio.

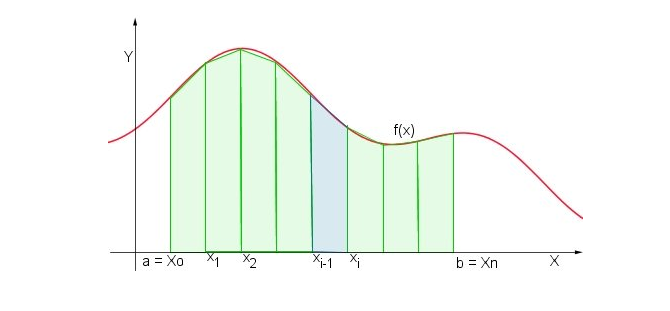


Si può approssimare se nell'intervallo di integrazione la funzione ha un andamento che si scosta poco dal lineare.

Se questo non accade si può suddividere l'intervallo complessivo  in un numero *n* opportuno di sotto intervalli quindi la regola del trapezio nella forma composta dice di applicare l'approssimazione precedente a tutti i sotto intervalli.

L’ampiezza di ogni intervallo è definita come segue:





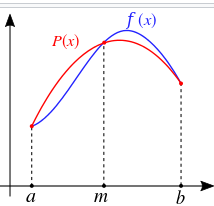
L’approssimazione dell’integrale definito è dato dalla somma dell’area di tutti i trapezi.

METODO SIMPSON

La formula di Simpson per l'integrazione numerica è simile a quella dei trapezi, ma anziché interpolare la funzione da integrare con i segmenti di retta per due punti, usa archi di parabola per tre punti. In tal modo si ha un notevole miglioramento della convergenza.



La formula richiede che l'intervallo [a, b] sia diviso in un numero pari di intervalli di ampiezza **h**, ed è ovviamente esatta per i polinomi di secondo grado, e un po' meno ovviamente esatta anche per quelli di terzo grado.



parziali e sommando i valori ottenuti sugli  intervalli parziali e per l'intero intervallo d'integrazione si ottiene il totale.

FUNZIONI UTILIZZATE

* MPI\_Init(&argc,&argv); : inizializza MPI.
* MPI\_Comm\_rank(MPI\_COMM\_WORLD, &myRank); : Ritorna il rank del processo chiamante
* MPI\_Comm\_size(MPI\_COMM\_WORLD, &nProc); : Ritorna il numero di processi disponibili
* MPI\_Reduce : Permette di semplificare operazioni matematiche/logiche,nel nostro caso ogni processo somma la propria parte e infine la invia al processo ricevente.
* MPI\_Barrier : Aspetta che ogni processo arrivi su questo punto prima di andare avanti.

Speedup e efficienza

Speedup =

Funzione usata : (pow(x,2)+ 3\*x)\*sin(x)

Metodo dei trapezi

|  |  |  |  |
| --- | --- | --- | --- |
| **Size:** | **1000000** | **10000000** | **100000000** |
| **1 Processo (seriale)** | 0.060314 s | 0.479304 s | 4.643264 s |
| **2 Processi** | 0.043921s | 0.246245 s | 2.363317 s |
| **4 Processi** | 0.019887 s | 0.137636 s | 1.190533 s |
| **8 Processi** | 0.006623 s | 0.086105 | 0.621964 |
| **Speed Up (2 Proc.)** | 1,37 | 1,94 | 1,96 |
| **Speed Up (4 Proc.)** | 3,03 | 3,48 | 3,90 |
| **Speed Up (8 Proc.)** | 9,10 | 5,56 | 7,46 |

Metodo di Simpson

|  |  |  |  |
| --- | --- | --- | --- |
| **Size:** | **1000000** | **10000000** | **100000000** |
| **1 Processo (seriale)** | 0.061788 s | 0.489213 s | 4.747122 s |
| **2 Processi** | 0.040077 s | 0.260519 s | 2.432776 s |
| **4 Processi** | 0.024641 s | 0.150318 s | 1.208422 s |
| **8 Processi** | 0.012690 s | 0.062616 s | 0.618821 |
| **Speed Up (2 Proc.)** | 1,54 | 1,87 | 1,95 |
| **Speed Up (4 Proc.)** | 2,50 | 3,25 | 3,92 |
| **Speed Up (8 Proc.)** | 4,86 | 7,81 | 7,67 |

Come lanciare il programma

mpirun -np num\_processori ./nomeProgrammaCompilato intervalli/numero trapezi a b funzione\_da\_integrare metodo\_di\_integrazione

Compilare con il comando:

mpicc integration\_213548\_defazio.c -o integration\_213548\_defazio -lm

Esempio con metodo simpson

mpirun -np 1 ./integration\_213548\_defazio 1000000 0 10 2 1