

ELABORATO 1 - INTEGRALE

LO BRUTTO FABIO / MAIONE PAOLO

DEFINIZIONE DEL PROBLEMA

Si vuole progettare un algoritmo in MPI per risolvere l'integrale definito tra a e b di una funzione $y = f(x)$ su p processori.

In particolare si utilizza l'infrastruttura S.C.o.P.E. per permettere l'esecuzione del software in un ambiente parallelo.

DESCRIZIONE DELL'ALGORITMO

In particolare le fasi dell'algoritmo, implementato nel file *eLaborato_1.c*, sono:

1) Divisione dell'intervallo $[a, b]$ in `num_intervalli` intervalli e distribuzione di questi ultimi in p processori: ognuno dei p processori calcola l'integrale sul sotto intervallo di

ampiezza $h = \frac{(b-a)}{\text{num_intervalli}}$ ricevuto dal processo *root* (cioè quello con rank 0);

2) Elaborazione dell'integrale parziale in parallelo con la **formula trapezoidale composta**

$$T_{n+1}(f) = \frac{h}{2} \left(f(a) + f(b) + 2 \sum_{i=1}^n f(a + ih) \right)$$

3) Combinazione dei risultati parziali nel processo *root* che determinerà il risultato finale.

A tal proposito sono state utilizzate le primitive fornite da MPI (rispettivamente per la prima fase `MPI_Scatterv()` e per la terza `MPI_Reduce()`).

Inoltre l'algoritmo progettato comprende anche il caso in cui il numero degli intervalli non è multiplo del numero di processori p a disposizione.

Si è scelto di misurare i tempi di esecuzione nel processo di rank 0 usando la primitiva `MPI_Wtime()` tra la fase 2 e la fase 3 scegliendo il minimo tra 3 misurazioni ripetute.

Infine, si osservi che i controlli di robustezza del software sono stati interamente delegati al processo *root*.

INPUT, OUTPUT E CONDIZIONI DI ERRORE

- ## ESEMPIO DI FUNZIONAMENTO

con $a=0$ e $b=1$ e con `num_intervalli = 100000`.

Esempio di funzionamento con 4 processori, in $[0,1]$ con 100000 intervalli

200 400 600 800 1000 1200 1400 1600 1800

ESEMPI DI ERRORE

Nelle successive immagini, invece, sono mostrati i messaggi di errore al verificarsi delle condizioni sopra citate.

```
%un esempio per ciascuna condizione di errore
errori
```

```

Error: numero di processori non positivo

[M63000769@ui-studenti ~/elaborati/elaborato_1/integrale]$ cat elaborato_1.out
-----
This job is allocated on 4 cpu(s)
Job is running on node(s):
wm273.scope.unina.it
wm273.scope.unina.it
wm273.scope.unina.it
wm273.scope.unina.it
-----
PBS: qsub is running on ui-studenti.scope.unina.it
PBS: originating queue is student1
PBS: executing queue is student1
PBS: working directory is /homes/DIS/CALCPAR/2019/M63000769/elaborati/elaborato_1/integrale
PBS: execution mode is PBS_BATCH
PBS: job identifier is 3920490.torque02.scope.unina.it
PBS: job name is elaborato_1
PBS: node file is /var/spool/pbs/aux//3920490.torque02.scope.unina.it
PBS: current home directory is /homes/DIS/CALCPAR/2019/M63000769
PBS: PATH = /usr/lib64/openmpi/1.2.7-gcc/bin:/usr/kerberos/bin:/opt/exp_soft/unina.it/intel/composer_xe_2013_spl.3.174/bin/intel64:/opt/exp_soft/unina.it/intel/composer_xe_2013_spl.3.174/mpirt/bin/intel64:/opt/exp_soft/unina.it/intel/composer_xe_2013_spl.3.174/bin/intel64:/opt/exp_soft/unina.it/intel/composer_xe_2013_spl.3.174/mic/bin/intel64_mic:/opt/exp_soft/unina.it/intel/composer_xe_2013_spl.3.174/debugger/ui/intel64:/opt/d-cache/srm/bin:/opt/d-cache/dcapi/bin:/opt/edg/bin:/opt/glibe/bin:/opt/globus/bin:/opt/lcg/bin:/usr/local/bin:/bin:/usr/bin:/opt/exp_soft/HADOOP/hadoop-1.0.3/bin:/opt/exp_soft/unina.it/intel/composerxe/bin/intel64:/opt/exp_soft/unina.it/MPXExpress/mpj-v0_38/bin:/homes/DIS/CALCPAR/2019/M63000769/bin
Esegui: /usr/lib64/openmpi/1.4-gcc/bin/mpicc -o /homes/DIS/CALCPAR/2019/M63000769/elaborati/elaborato_1/integrale/elaborato_1 /homes/DIS/CALCPAR/2019/M63000769/elaborati/elaborato_1/integrale/elaborato_1.c
Esegui: /usr/lib64/openmpi/1.4-gcc/bin/mpixec -machinefile /var/spool/pbs/aux//3920490.torque02.scope.unina.it -np 4 /homes/DIS/CALCPAR/2019/M63000769/elaborati/elaborato_1/integrale/elaborato_1
Errore! Il numero dei processi deve essere maggiore di zero
[M63000769@ui-studenti integrale]$
[M63000769@ui-studenti integrale]$
[M63000769@ui-studenti integrale]$
[M63000769@ui-studenti integrale]$
[M63000769@ui-studenti integrale]$
[M63000769@ui-studenti integrale]$

```

100
200
300
400
500
600
700
800
900
1000

1000

100
200
300
400
500
600
700
800
900
1000

1000

100
200
300
400
500
600
700
800
900
1000

1000

100
200
300
400
500
600
700
800
900
1000

1000

ANALISI DELLE PRESTAZIONI (T(p), S(p), E(p))

Di seguito per brevità si indicherà con p il numero di processori e con N il numero di intervalli.

Tempo di esecuzione - $T(p)$

Si è scelto di misurare i tempi di esecuzione nel processo *root* usando la primitiva `MPI_Wtime()`. In particolare l'intervallo di tempo misurato è quello che comprende le fasi 2 e 3 dell'algoritmo prima citate.

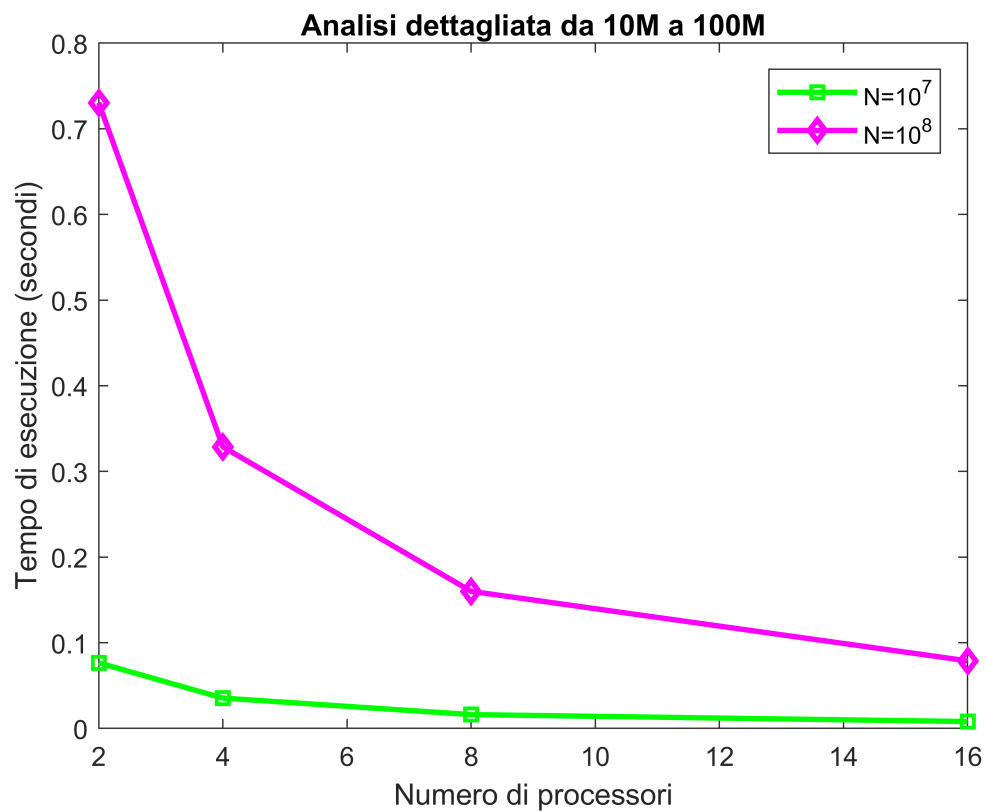
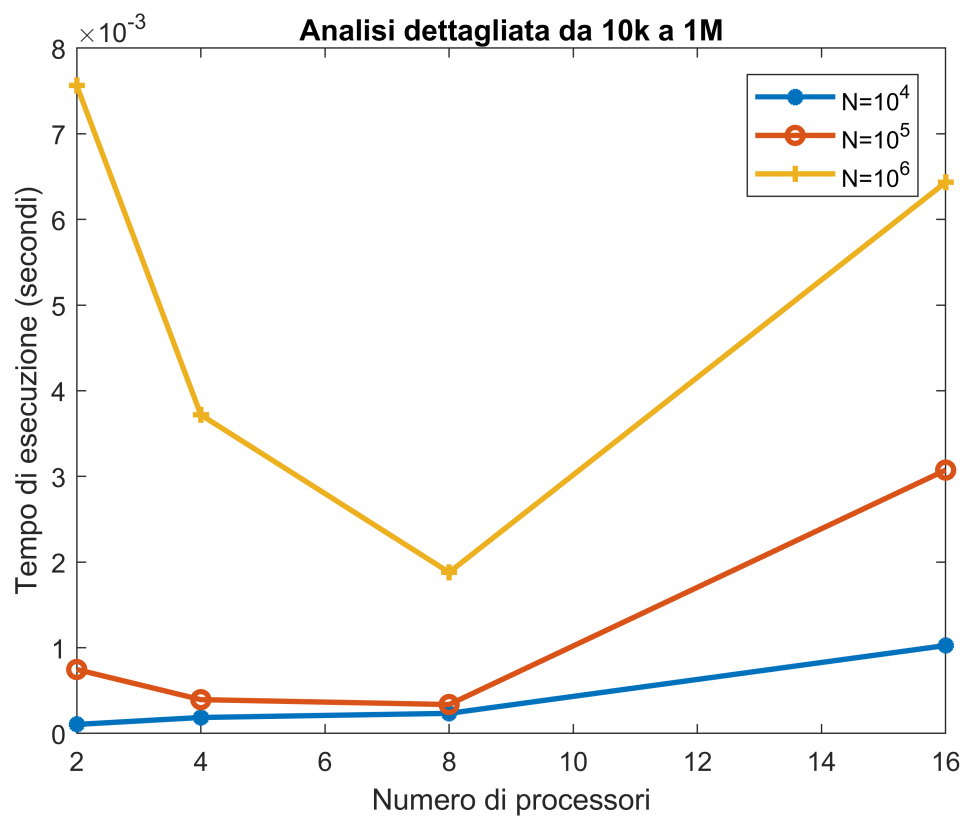
Per ciascuna misurazione (al variare di N da 10k a 100M e al variare di p da 2 a 16) è stato considerato il minimo tra 3 esecuzioni ripetute, eseguite in momenti diversi.

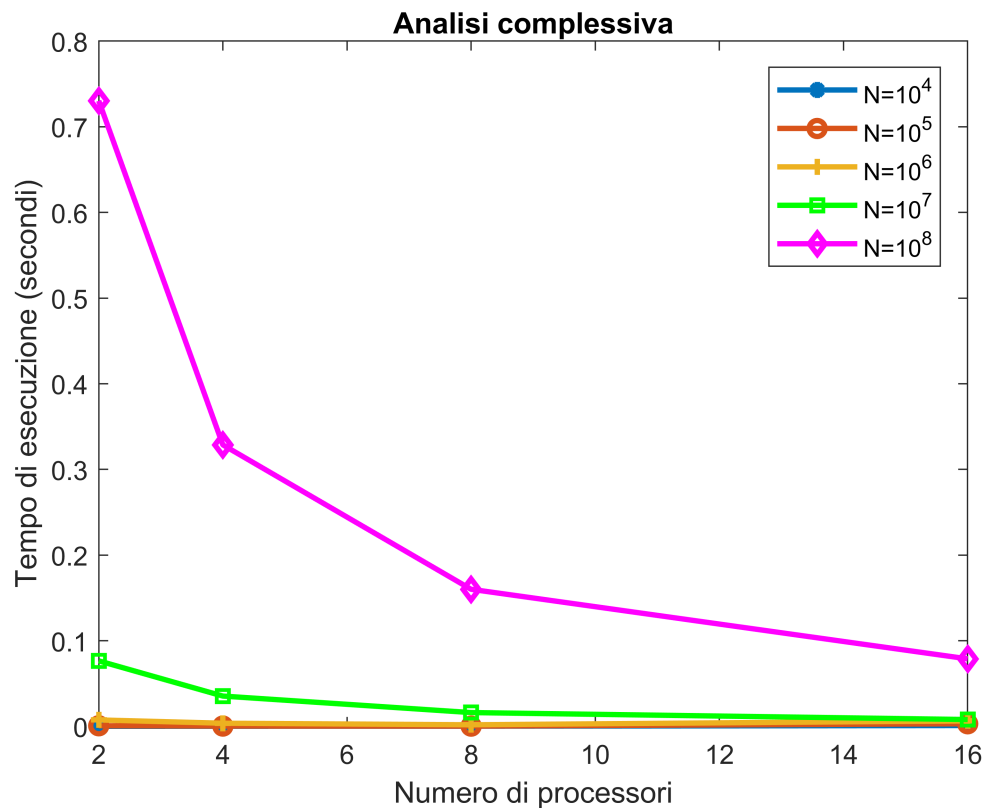
Di seguito si riportano i risultati in forma di tabelle e grafici.

```
%esecuzione script per tabelle e grafici  
tempi
```

Warning: Image is too big to fit on screen; displaying at 67%

Tempi					
TEMPI	10000	100000	1000000	10000000	100000000
2	0,0001049041748047	0,000746011734009	0,00756308097839	0,0765459632873535	0,730383014678955
4	0,0001859664916992	0,000394105911255	0,00371885299683	0,0354251861572266	0,328655014038085
8	0,0002341270446777	0,000337123870850	0,00187993049622	0,0161489322662354	0,160068035125732
16	0,0010261535644531	0,003073930740356	0,00643301010132	0,0080289840698242	0,078861236572266





L'ultimo grafico è quello che riassume i risultati ottenuti per tutti i possibili valori di N e p . Sono forniti due ulteriori grafici (i primi due) che mostrano gli andamenti più nel dettaglio.

Per considerazioni più di dettaglio su questi risultati si rimanda alla sezione Conclusioni.

Speed up ed Efficienza - $S(p)$ ed $E(p)$

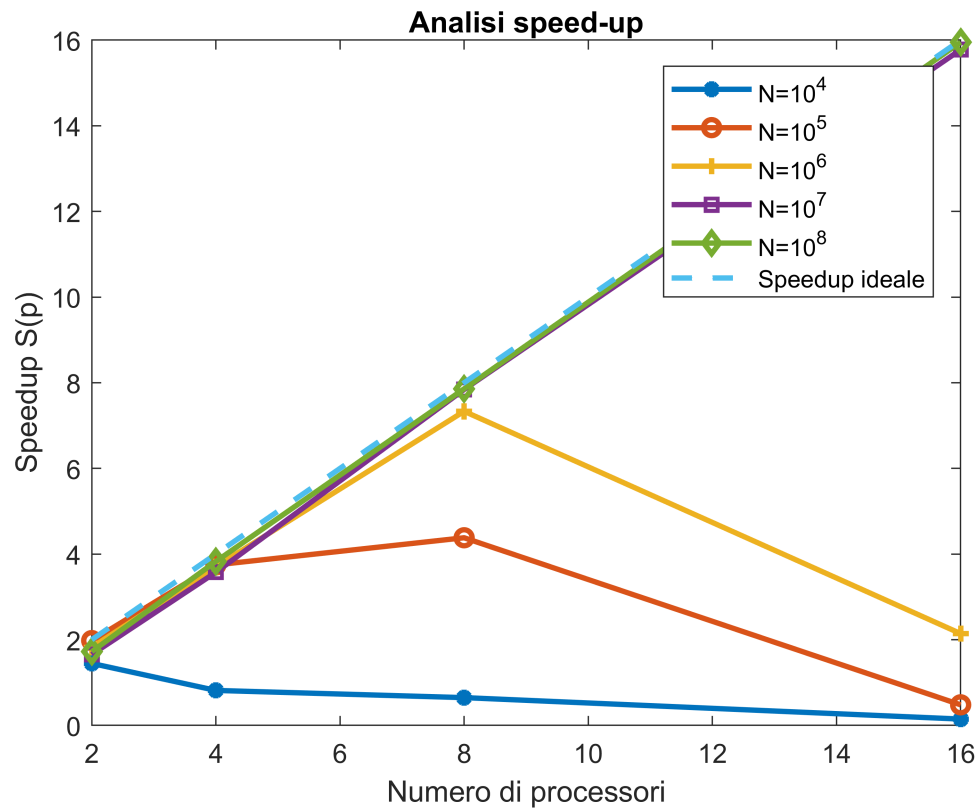
Si è calcolato, inoltre, il tempo di riferimento $T(1)$ che corrisponde al tempo di esecuzione su un unico processore.

A partire dai tempi misurati nella sezione precedente e da $T(1)$ è stato calcolato lo speed-up al variare di N e p .

```
%esecuzione script per tabelle e grafici
speedup
```

Warning: Image is too big to fit on screen; displaying at 67%

Speed-up					
SPEEDUP	10000	100000	1000000	10000000	100000000
2	1,4477	1,9799	1,8229	1,6548	1,7224
4	0,8167	3,7477	3,7073	3,5756	3,8277
8	0,6487	4,3812	7,3338	7,8437	7,8590
16	0,1480	0,4805	2,1432	15,7763	15,9518

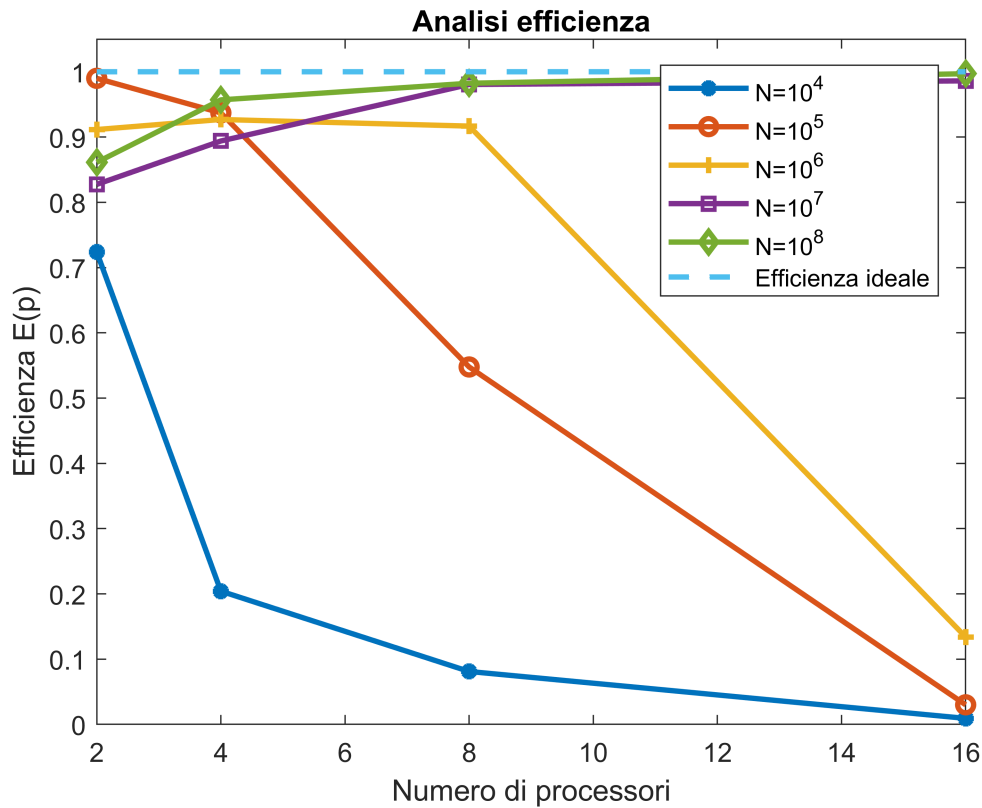


Infine si è calcolata l'efficienza rapportando lo speed up $S(p)$ al numero di processori p .

```
%esecuzione script per tabelle e grafici
efficienza
```

Warning: Image is too big to fit on screen; displaying at 67%

Efficienza					
EFFICIENZA	10000	100000	1000000	10000000	100000000
2	0,7239	0,9899	0,9115	0,8274	0,8612
4	0,2042	0,9369	0,9268	0,8939	0,9569
8	0,0811	0,5476	0,9167	0,9805	0,9824
16	0,0093	0,0300	0,1339	0,9860	0,9970



Conclusioni

Dai grafici appena presentati si possono trarre alcune considerazioni.

Analizzando l'efficienza si nota come nel caso $N=10^4$ e $N=10^5$ l'efficienza ottimale si ha per 2 processori.

Invece nel caso di $N=10^6$ l'ottimo è in corrispondenza di 4 processori, mentre per $N=10^7$ e $N=10^8$ è 16 processori. Ciò può portare a dire che con molta probabilità l'ottimo per 8 processori si trova in corrispondenza di un valore intermedio tra $N=10^6$ e $N=10^7$.

Si possono fare ulteriori considerazioni notando che, per N fissato, l'efficienza peggiora dopo un certo valore di p (ciò verifica sperimentalmente la legge di Amdahl), e che, in generale, all'aumentare sia di N che di p , l'efficienza migliora (verificando la legge di Gustafson).

Analoghe considerazioni per i tempi e lo speedup.

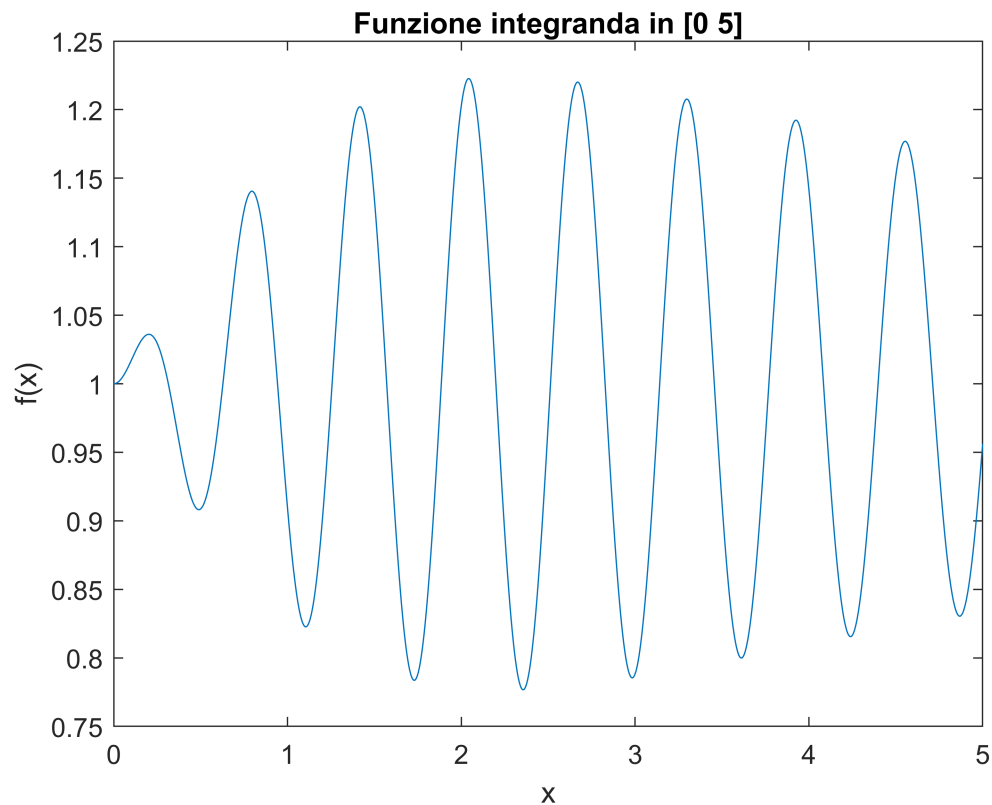
ANALISI DELL' ACCURATEZZA

Confrontando i risultati ottenuti sul cluster Scope e quelli ottenuti su MATLAB si ottengono i seguenti errori relativi. I livelli di accuratezza sono stati misurati fissando a 8 il numero di processori, e facendo variare la dimensione del problema num_intervalli. Gli estremi di integrazione scelti sono $a=0$, $b=5$.

Per i test è stata usata la seguente funzione integranda:

$$f(x) = 1 + \sin(10x) * \frac{x}{x^2 + 5}$$

```
%grafico della funzione integranda  
grafico
```



```
%esecuzione script per i test di accuratezza  
accuratezza
```

```
F = function_handle with value:  
@(x)1+sin(10.*x).*(x./(x.^2+5))  
risultato_matlab =  
    4.983979125364205e+00  
err_rel_10k_intervalli =  
    9.594707130727628e-05  
err_rel_100k_intervalli =  
    9.599362810692822e-06
```

```
err_rel_1M_intervalli =  
    9.651938468569100e-07  
err_rel_10M_intervalli =  
    1.017821889124140e-07  
err_rel_100M_intervalli =  
    1.540213007827612e-08
```