

# 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

- **Input:** la funzione  $f$  da integrare, i due estremi dell'intervallo  $a$  e  $b$ , il numero di intervalli `num_intervalli` e il numero di processori (`numero_processori`) da utilizzare.
- **Output:** l'approssimazione dell'integrale definito tra  $a$  e  $b$  di  $f$ .
- **Condizioni di errore:** l'estremo  $b$  non deve essere minore di  $a$ , il numero di intervalli non deve essere minore del numero di processori specificato in ingresso. In particolare quest'ultimo deve coincidere con il corrispondente parametro `nodes` del file di configurazione `eLaborato_1.pbs`.

## ESEMPIO DI FUNZIONAMENTO

Nell'immagine seguente vi è un esempio di funzionamento con  $f(x) = \frac{x}{x^2 + 5}$ ,

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

%esempio di funzionamento  
funzionamento

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

```
[M6300076@ui-studenti integrale]$ cat elaborato_1.out
This job is allocated on 4 cpu(s)
Job is running on node(s):
wn273.scope.unina.it
wn273.scope.unina.it
wn273.scope.unina.it
wn273.scope.unina.it
-----
PBS: qsub is running on ui-studenti.scope.unina.it
PBS: originating queue is studenti
PBS: executing queue is studenti
PBS: working directory is /homes/DIS/CALCPAR/2019/M63000769/elaborati/elaborato_1/integrale
PBS: execution mode is PBS_BATCH
PBS: job identifier is 3917729.torque02.scope.unina.it
PBS: job name is elaborato_1
PBS: node file is /var/spool/pbs/aux//3917729.torque02.scope.unina.it
PBS: current home directory is /homes/DIS/CALCPAR/2019/M63000769
PBS: PATH = /usr/lib64/openssl/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/mpi/bin/intel64:/opt/exp_soft/unina.it/intel/composer_xe_2013_spl.3.174/bin/intel
64:/opt/exp_soft/unina.it/intel/composer_xe_2013_spl.3.174/bin/intel64_mic:/opt/exp_soft/unina.it/intel/composer_xe_2013_spl.3.174/debu
gger/gui/intel64:/opt/d-cache/srm/bin:/opt/d-cache/dcap/bin:/opt/edg/bin:/opt/glite/bin:/opt/globus/bin:/opt/lcg/bin:/usr/local/bin:/bi
n:/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/MPJExpress
/mpj-v0_38/bin:/homes/DIS/CALCPAR/2019/M63000769/bin
-----
$eguo: /usr/lib64/openssl/1.4-gcc/bin/mpicc -o /homes/DIS/CALCPAR/2019/M63000769/elaborati/elaborato_1/integrale/elaborato_1 /homes/DI
S/CALCPAR/2019/M63000769/elaborati/elaborato_1/integrale/elaborato_1.c
$eguo: /usr/lib64/openssl/1.4-gcc/bin/mpixec -machinefile /var/spool/pbs/aux//3917729.torque02.scope.unina.it -np 4 /homes/DIS/CALCPA
R/2019/M63000769/elaborati/elaborato_1/integrale/elaborato_1
Ciao sono il processo 1 e mi chiamo wn273.scope.unina.it
Ciao sono il processo 2 e mi chiamo wn273.scope.unina.it
Ciao sono il processo 3 e mi chiamo wn273.scope.unina.it
Ciao sono il processo 0 e mi chiamo wn273.scope.unina.it
Sono il processo 0. Il mio integrale parziale è: 0.0062117538360116
Sono il processo 1. Il mio integrale parziale è: 0.0181842806380498
Sono il processo 2. Il mio integrale parziale è: 0.0289101813763584
Sono il processo 3. Il mio integrale parziale è: 0.0378562292124507
Sono il processo 0. L'integrale è pari a : 0.0911624450628705.
Sono il processo 0 : tempo di esecuzione totale di 0.0007870197296143 secondi.
[M6300076@ui-studenti integrale]$
[M6300076@ui-studenti integrale]$
```

## 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

### Errore: numero di processori diverso da quello specificato

```
[M63000769@ui-studenti integrale]$ cat elaborato_1.out
-----
This job is allocated on 8 cpu(s)
Job is running on node(s):
wm273.scope.unina.it
wm273.scope.unina.it
wm273.scope.unina.it
wm273.scope.unina.it
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 studenti
PBS: executing queue is studenti
PBS: working directory is /homes/DIS/CALCPAR/2019/M63000769/elaborati/elaborato_1/integrale
PBS: execution mode is PBS_BATCH
PBS: job identifier is 3917718.torque02.scope.unina.it
PBS: job name is elaborato_1
PBS: node file is /var/spool/pbs/aux//3917718.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_sp1.3.174/bin/intel64:/opt/exp_soft/unina.it/intel/composer_xe_2013_sp1.3.174/mpiirt/bin/intel64:/opt/exp_soft/unina.it/intel/composer_xe_2013_sp1.3.174/bin/intel64:/opt/exp_soft/unina.it/intel/composer_xe_2013_sp1.3.174/bin/intel64_mic:/opt/exp_soft/unina.it/intel/composer_xe_2013_sp1.3.174/debugger/intel64:/opt/d-cache/srm/bin:/opt/d-cache/dcap/bin:/opt/edg/bin:/opt/glite/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/MPExpress/bin:/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/mpiexec -machinefile /var/spool/pbs/aux//3917718.torque02.scope.unina.it -np 8 /homes/DIS/CALCPAR/2019/M63000769/elaborati/elaborato_1/integrale/elaborato_1
Errore! Il numero dei processi deve essere pari a 4
[M63000769@ui-studenti integrale]$
[M63000769@ui-studenti integrale]$
```

100  
200  
300  
400  
500  
600  
700  
800  
900

200      400      600      800      1000      1200      1400      1600      1800

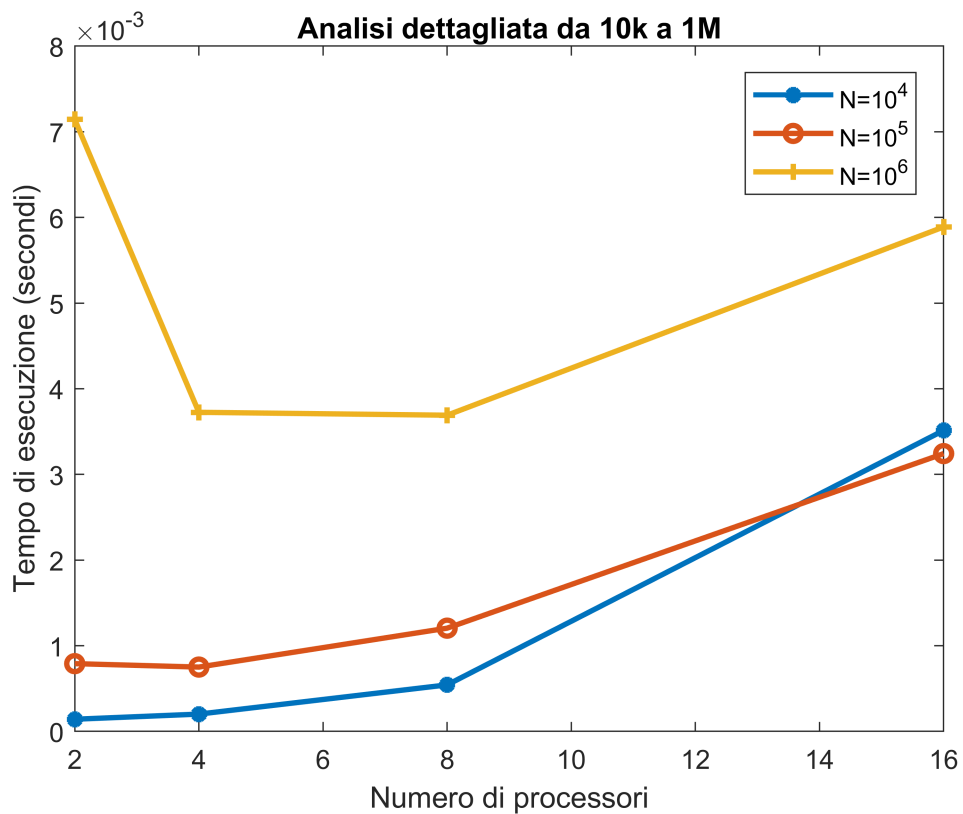
100  
200  
300  
400  
500  
600  
700  
800  
900

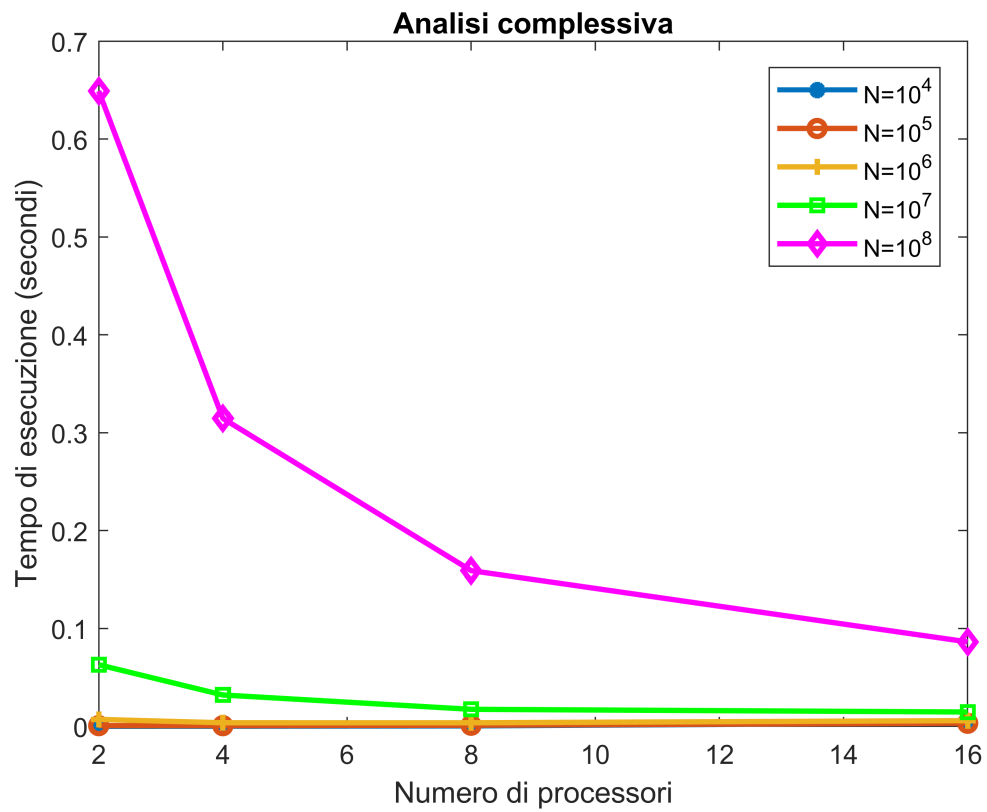
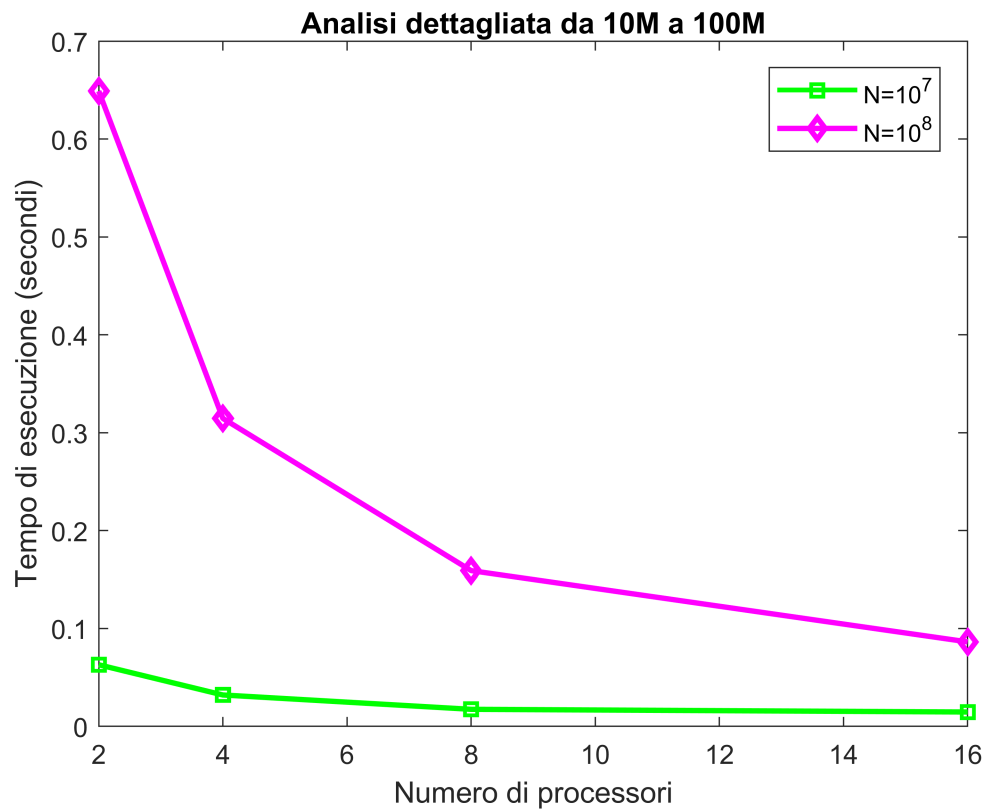
200      400      600      800      1000      1200      1400      1600      1800

100  
200  
300  
400  
500  
600  
700  
800  
900

200      400      600      800      1000      1200      1400      1600      1800

Tempi				
	x100000	x1000000	x10000000	x100000000
2	7.910728454590000e-04	7.143974304199200e-03	6.288814544677730e-02	6.274521350860590e-01
4	7.510185241699000e-04	3.725051879882800e-03	3.204798698425290e-02	3.145918846130370e-01
8	1.205205917358400e-03	3.691196441650400e-03	1.745796203613280e-02	1.591570377349850e-01
16	3.243207931518600e-03	5.888938903808600e-03	1.471495628356930e-02	8.632898330688480e-02





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.

Tali risultati verificano la legge di Amdahl: all'aumentare del numero di processori, fissata la dimensione del problema (N), i tempi peggiorano (cioè aumentano). Per esempio, si osservi il comportamento nel caso di  $N = 10^6$  nel grafico "Analisi dettagliata da 10k a 1M". Inoltre è verificata anche la legge di Gustafson in quanto all'aumentare sia di N che di p, il tempo di esecuzione migliora (cioè diminuisce) come è evidente nel grafico "Analisi complessiva".

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

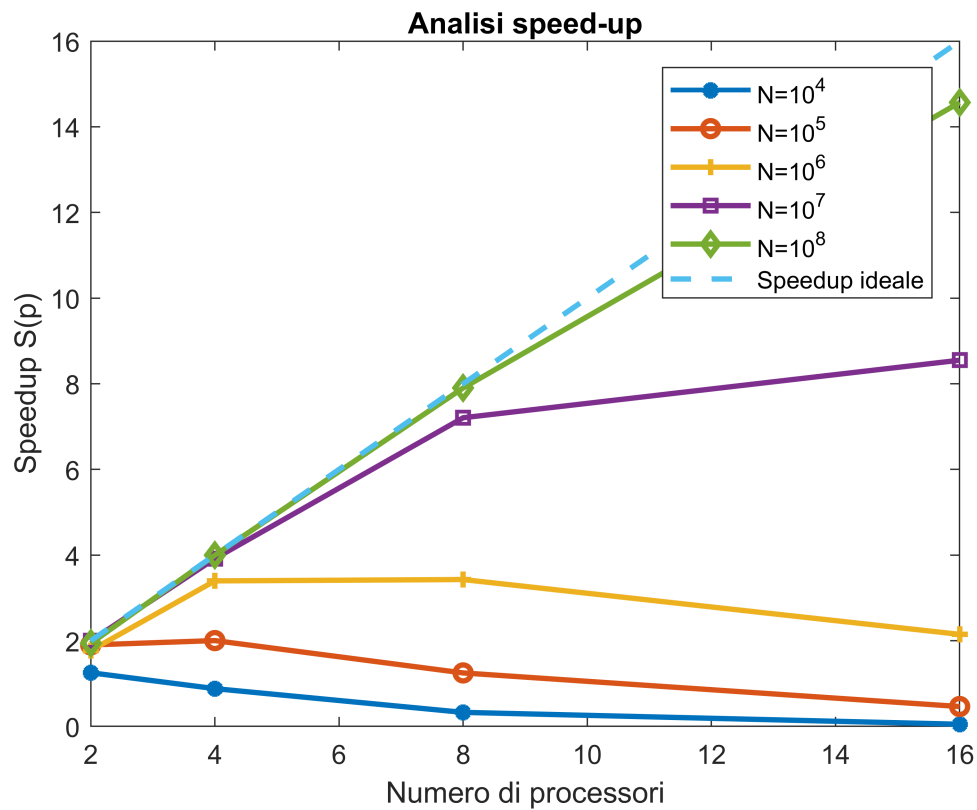
Si è calcolato, inoltre, il tempo di riferimento T(1) che corrisponde al tempo di esecuzione su un unico processore.

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

		Speed-up								
		x10000	x100000	x1000000	x10000000	x100000000				
50	2	1.2534	1.9024	1.7716	1.9978	1.9379				
100	4	0.8819	2.0038	3.3975	3.9183	3.9975				
150	8	0.3279	1.2487	3.4287	7.2066	7.9017				
	16	0.0507	0.4640	2.1491	8.5499	14.5677				
		100	200	300	400	500	600	700	800	900

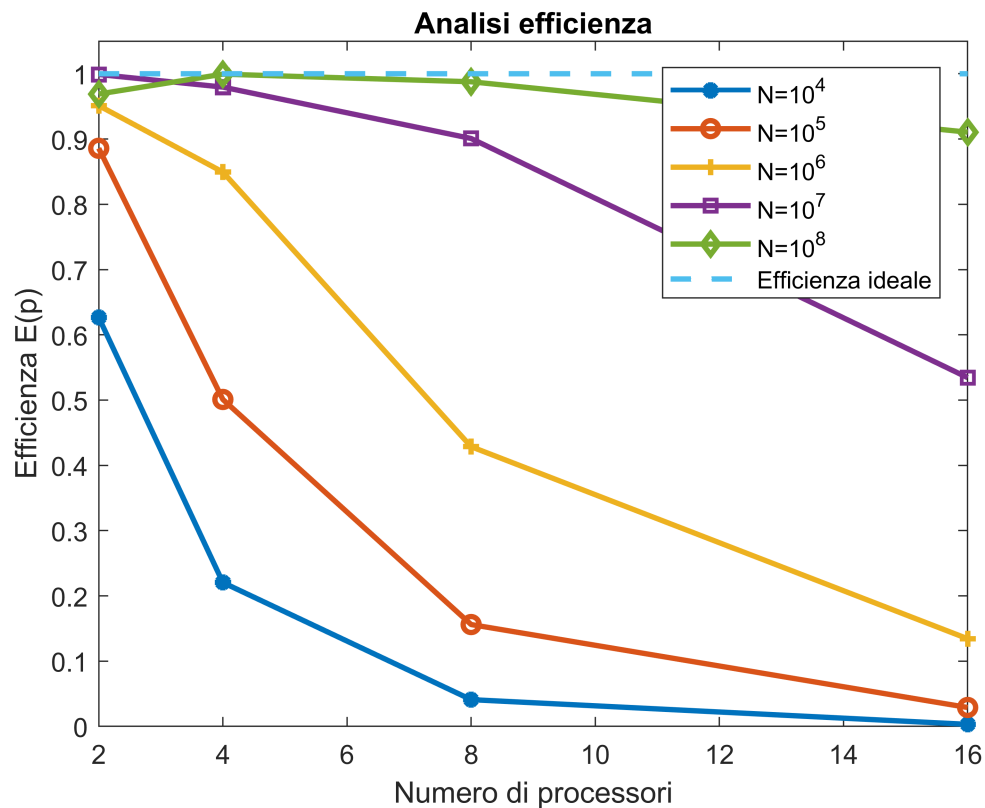




Infine si è calcolata l'efficienza rapportando lo speed up  $S(p)$  al numero di processori  $p$ .

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

Efficienza										
	x10000	x100000	x1000000	x10000000	x100000000					
50	2	0.6267	0.8858	0.9512	0.9989	0.9689				
100	4	0.2205	0.5010	0.8494	0.9796	0.9994				
150	8	0.0410	0.1561	0.4286	0.9008	0.9877				
	16	0.0032	0.0290	0.1343	0.5344	0.9105				
		100	200	300	400	500	600	700	800	900



## Conclusioni

Dai grafici appena presentati si possono trarre alcune considerazioni.

Analizzando l'efficienza si nota come nel caso  $N=10^4$  l'efficienza è bassa anche per 2 processori; ciò indica che è un problema che non conviene risolvere in parallelo, fissate queste dimensioni.

Tuttavia anche da  $N=10^5$  fino a  $N=10^7$  si osserva che l'efficienza migliore si ottiene in corrispondenza di soltanto 2 processori.

Si possono ripetere analoghe considerazioni come quelle fatte per i tempi, notando che, per  $N$  fissato, l'efficienza peggiora dopo un certo valore di  $p$ , e che, in generale, all'aumentare sia di  $N$  che di  $p$ , l'efficienza migliora.

Solo nel caso in cui  $N=10^8$  l'efficienza ottima si ha con  $p=4$  processori.

Analoghe considerazioni per lo speedup.

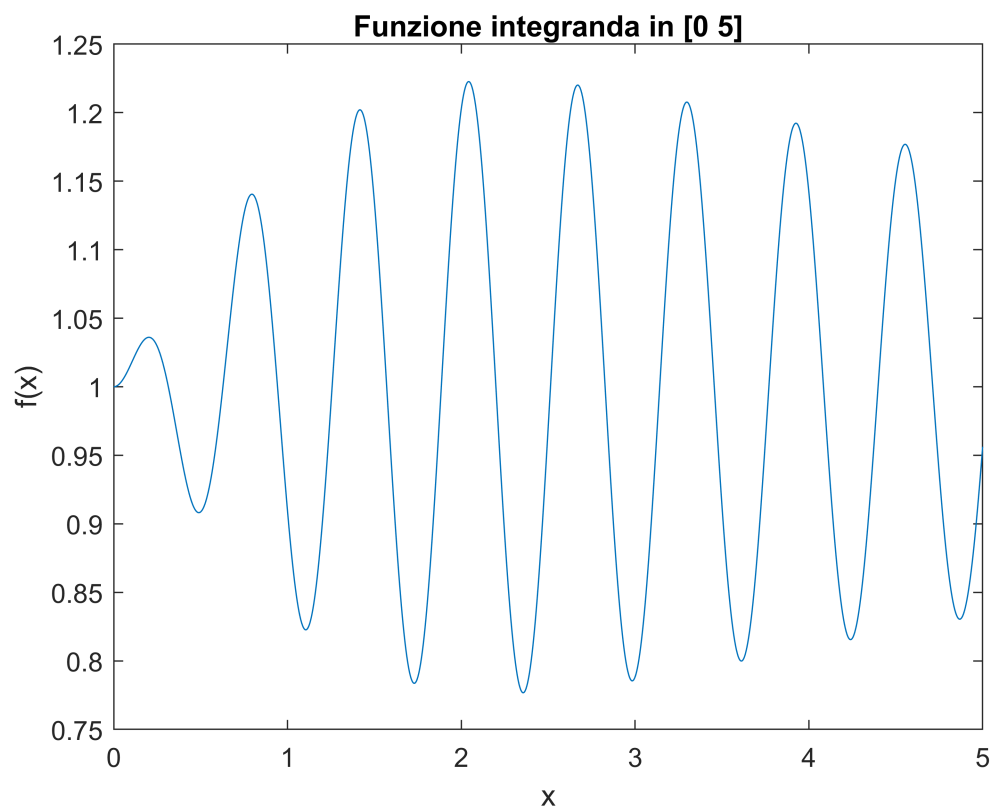
## ANALISI DELL' ACCURATEZZA

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 =  
    9.594707130727628e-05  
err_rel_100k =
```

```
9.599362810692822e-06
err_rel_1M =
9.651938468569100e-07
err_rel_10M =
1.017821889124140e-07
err_rel_100M =
1.540213007827612e-08
```

Ovviamente si può constatare che all'aumentare di num\_intervalli l'accuratezza dell'approssimazione migliora, ossia l'algoritmo converge.