

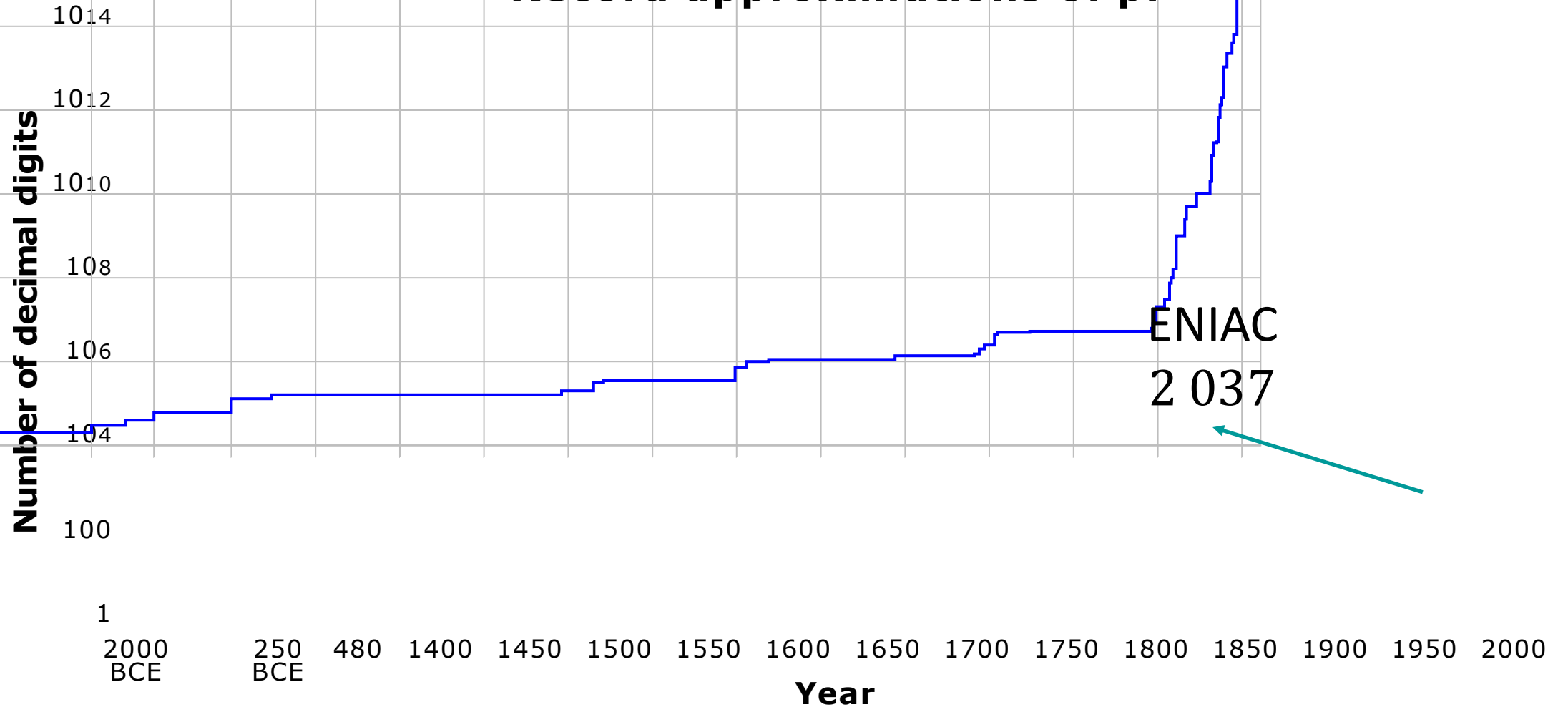
Sistemas Embarcados (INF01059)

2º Trabalho Prático

ANDREWS BERNI	167130
LETÍCIA DOS SANTOS	275604
IZADORA DOURADO BERTI	275606

Problema: Cálculo de pi

Record approximations of pi



Fonte: https://pt.wikipedia.org/wiki/Cronologia_do_c%C3%A1lculo_de_%CF%80

Métodos utilizados

- Simpson.
- Montecarlo.
- Soma de retângulos (Integral).
- Variável n em comum.



Linguagens

- Python 3.7.5
- Linguagem C

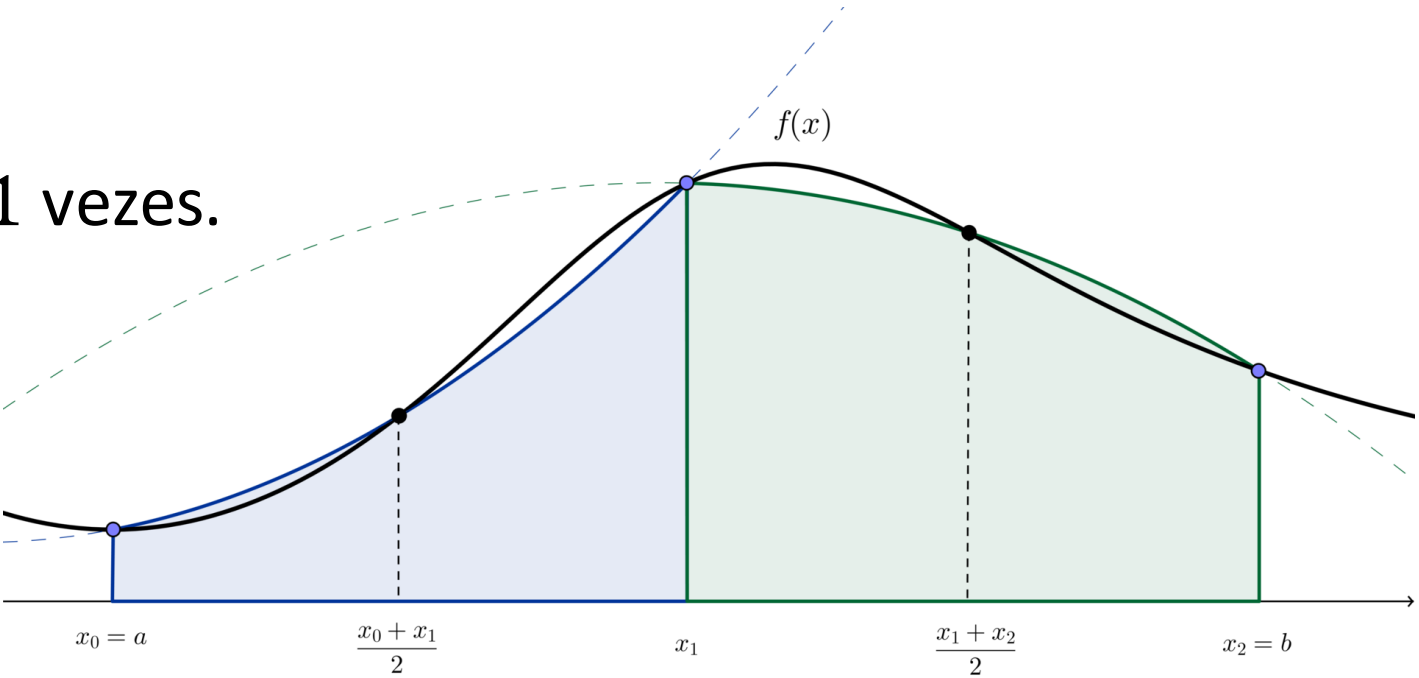
$h = (b - a)/n$

$n) \Big] \cdot$

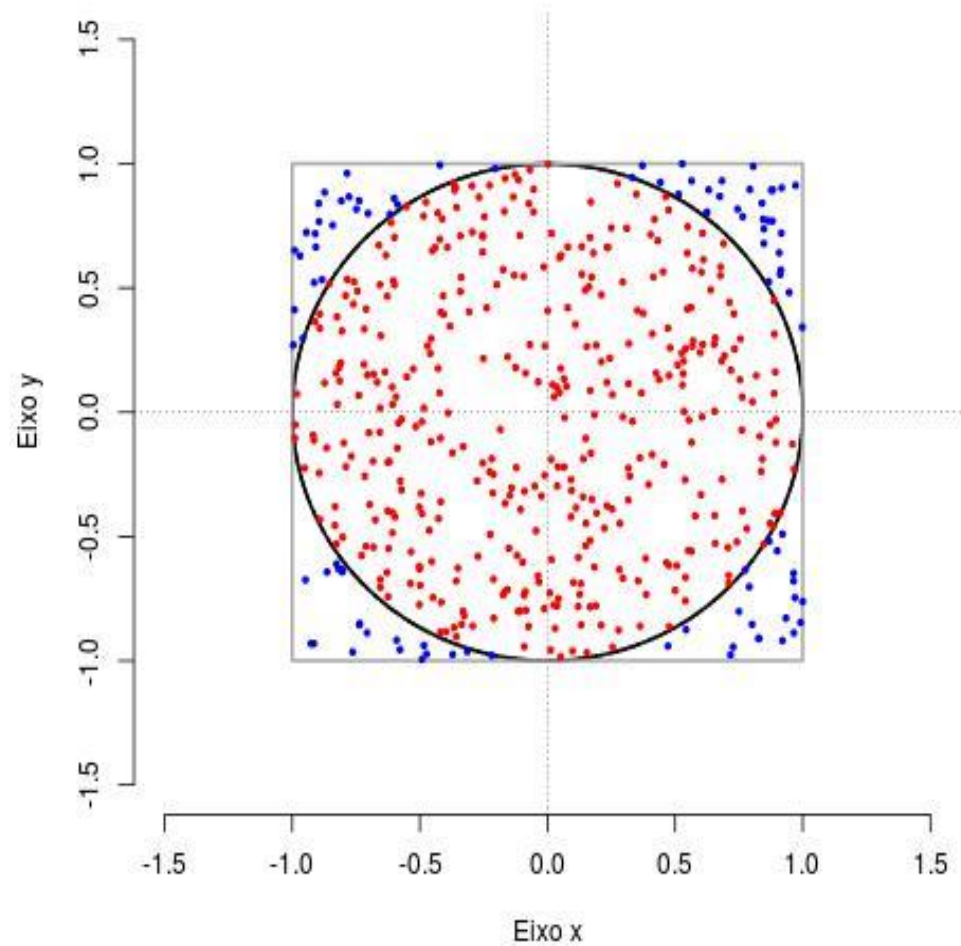
Simpson

A função será calculada $n + 1$ vezes.

$$\int_0^1 \frac{4}{1 + x^2} dx$$



Montecarlo



- Calculo de n pontos.

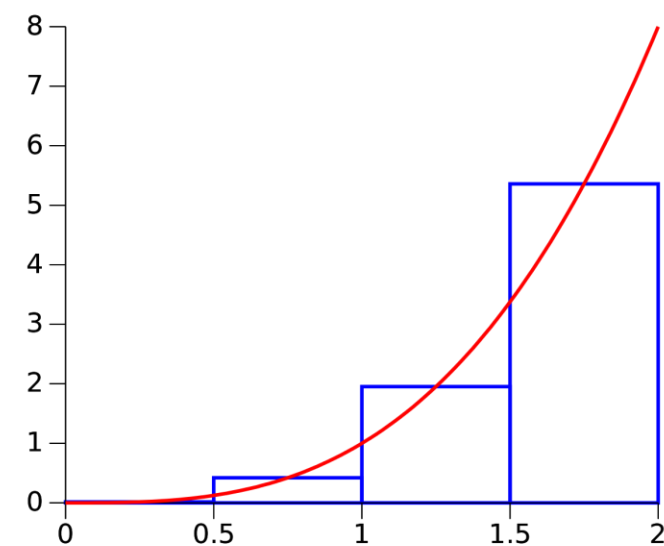
- $$pi = 4 * \frac{count}{n}$$

$$S = \sum_{i=1}^n f(x_i^*) \Delta x_i$$

Soma de retângulos

- $\int_a^b f(x) dx = S$

- n retângulos.



Máquina de teste

- Intel® Core™ i7-6700
- Número de núcleos = 4
- Nº de threads = 8
- Frequência baseada em processador 3.40 GHz
- gcc (Ubuntu 7.4.0-1ubuntu1~18.04.1) 7.4.0

Resultados

```
1 threads: SERIAL-----  
  
SIMPSON  
Introduce the number of nodes:  
1000000  
Energia consumida em Joules: 0.13892  
Approximated pi value: 3.141592653589788  
Total execution time: 0.0177649399847724 sec  
  
MONTECARLO  
Introduce the number of iterations n:  
1000000  
Energia consumida em Joules: 0.19885  
Approximate pi value: 3.1415120000000000821671619633  
Final execution time: 0.0270153420278803 sec  
  
MONTECARLOpyhton  
1000000  
3.141288  
Energia: 4.278859 J  
Potencia: 17.15296669112017  
tempo final: 0.24910306930541992  
  
INTEGRAL  
Introduce the number of iterations n:  
1000000  
pi eh 3.141593  
Energia consumida em Joules: 0.18439  
Tempo: 0.012623  
  
8 threads: PARALELO-----  
  
INTEGRAL  
Introduce the number of iterations n:  
1000000  
pi eh 3.141593  
Energia consumida em Joules: 0.11279  
Tempo: 0.004264
```

Resultados para $n = 1\,000\,000$

Algoritmo	Tempo (em s)	Energia (em J)
Simpson	0,017765	0,13892
Montecarlo em C	0,027015	0,19885
Montecarlo em Python	0,249103	4,27886
Soma	0,012623	0,18439
Soma paralela em 8 threads OMP	0,004264	0,11279

Tempo (em s)

Soma paralela em 8 threads OMP

Soma

Montecarlo em Python

Montecarlo em C

Simpson

0,00

0,05

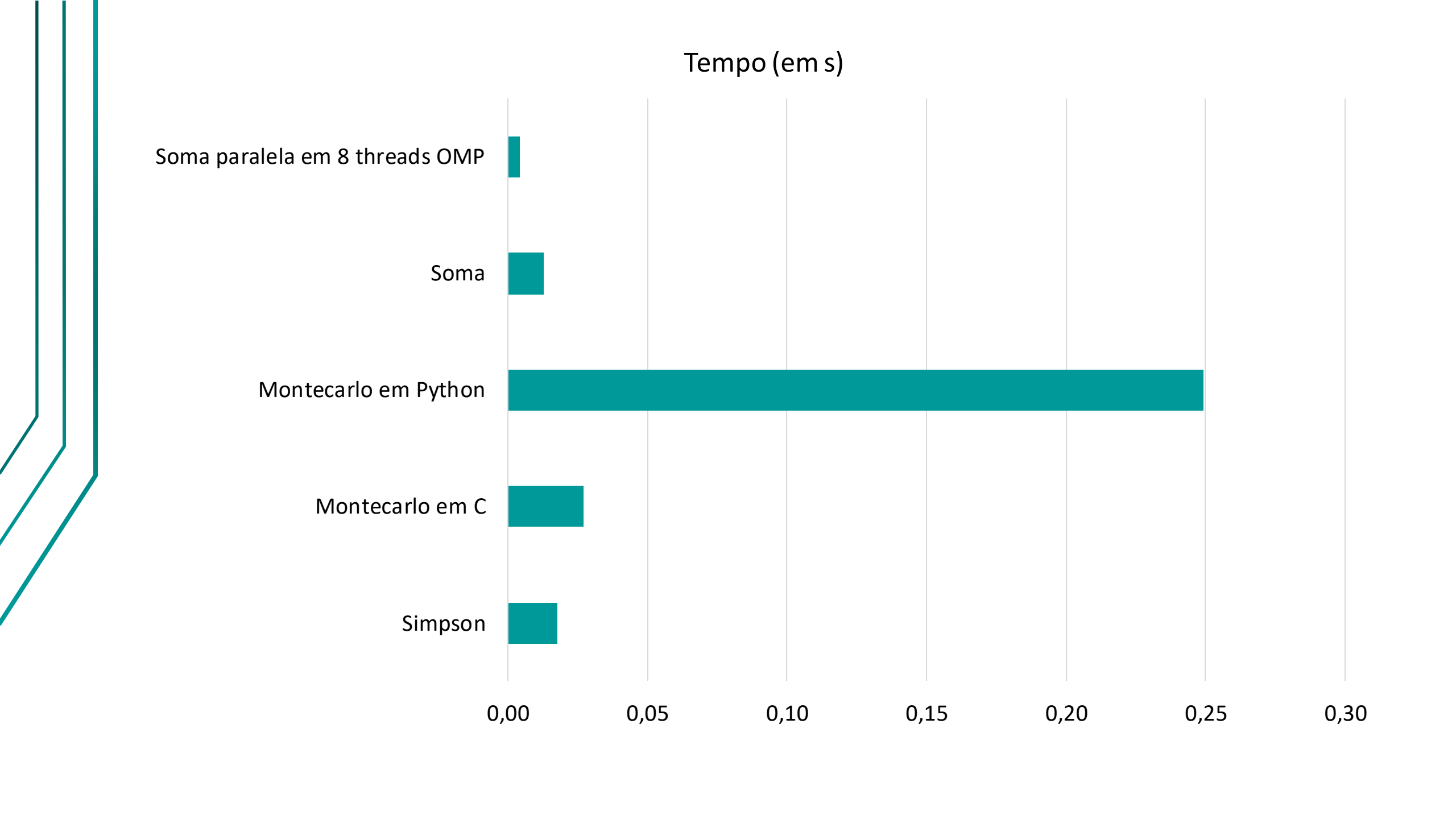
0,10

0,15

0,20

0,25

0,30



Tempo (em s)

Soma paralela em 8 threads OMP

Soma

Montecarlo em C

Simpson

0,000

0,005

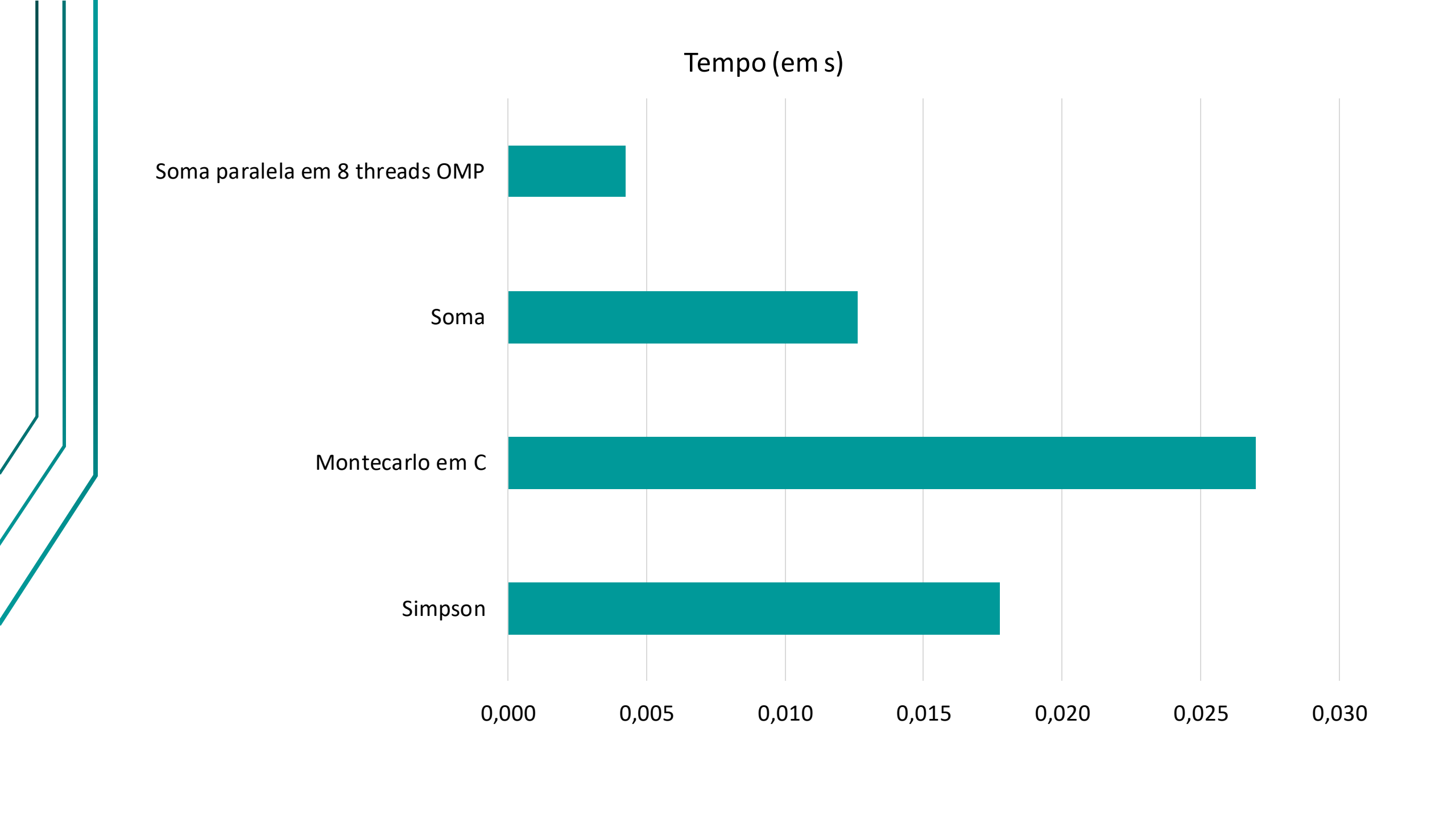
0,010

0,015

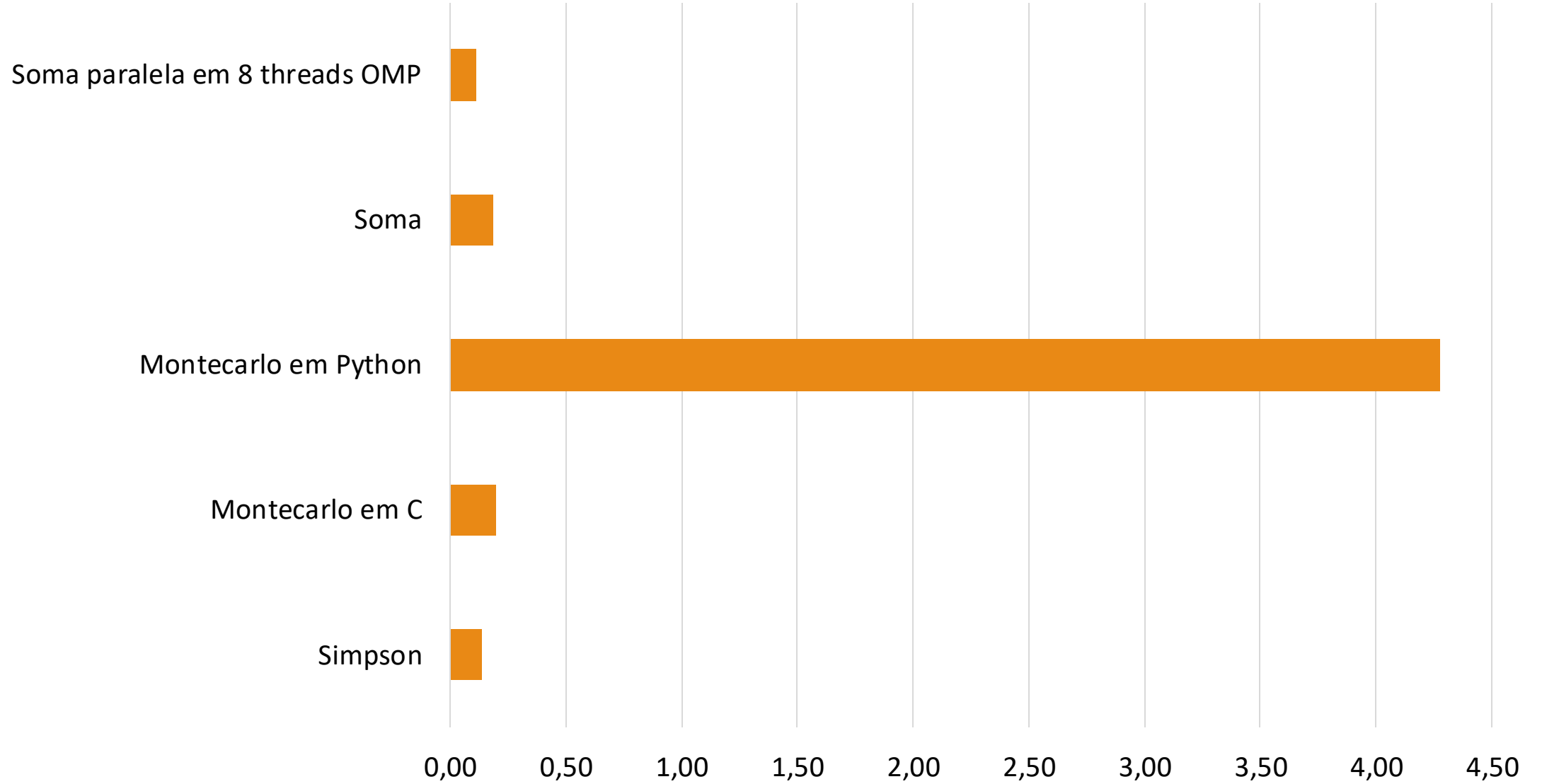
0,020

0,025

0,030



Energia (em J)



Energia (em J)

Soma paralela em 8 threads OMP

Soma

Montecarlo em C

Simpson

0,00

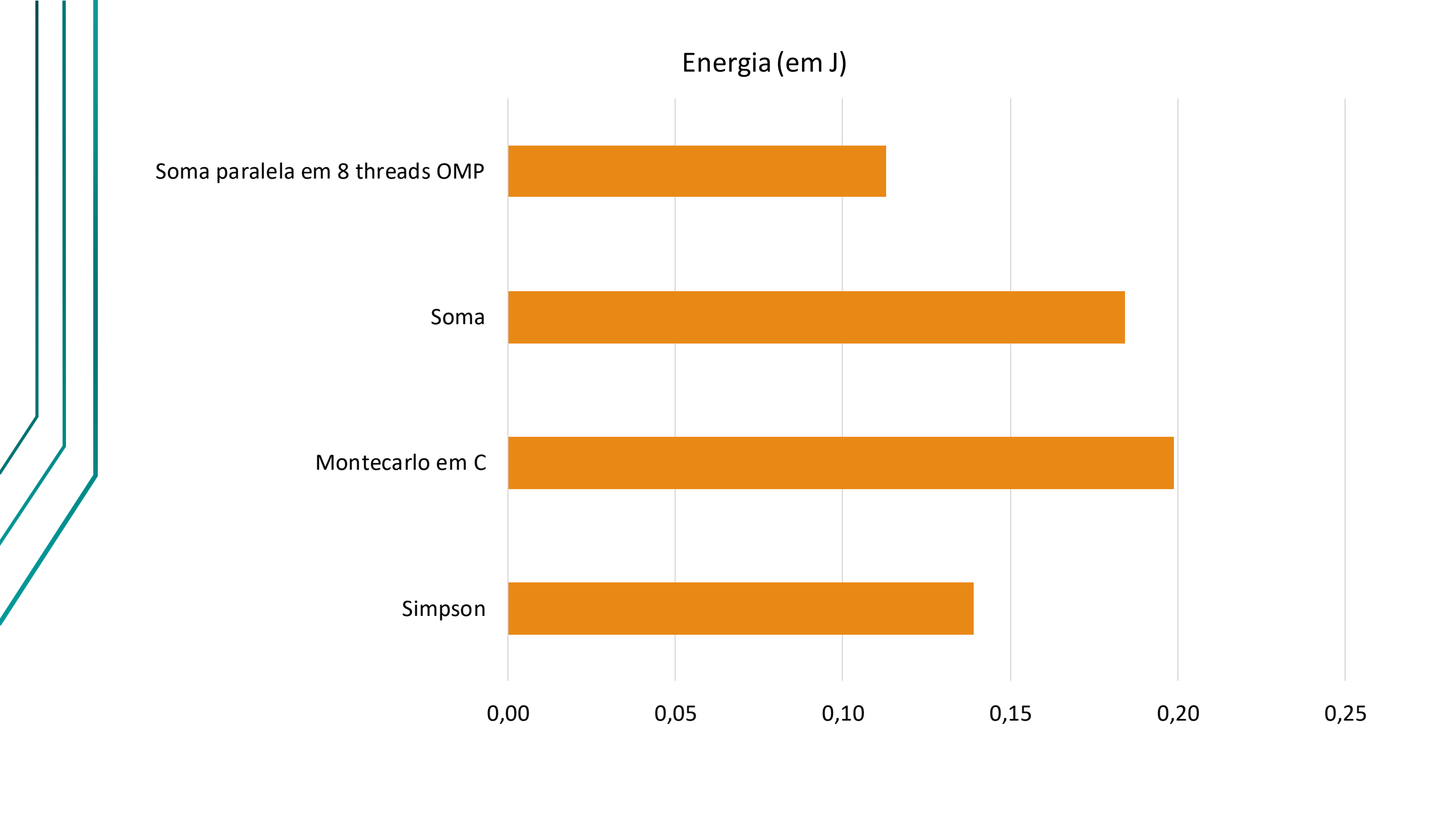
0,05

0,10

0,15

0,20

0,25



Conclusões: Versão Python e C

- Quando compilamos e linkamos um programa em linguagens C e C++, o programa inteiro é convertido em instruções executáveis nativa.
- Python é uma linguagem de programação interpretada e dinâmica baseada em bytecode que é executada através da máquina virtual.
- Como visto no início da cadeira e conseguimos analisar nos gráfico anteriores, tempo de execução médio do algoritmo de MonteCarlo em c é 10x mais rápido do que na linguagem em Python.
- Mas não é a mesma proporção da energia. A linguagem C consome cerca de 20x menos energia para o algoritmo sequencial MonteCarlo.
- Essa diferença de energia não é proporcional ao tempo de execução, pois o que impacta no consumo de energia é o uso de instrução no processador

Para $n = 100\ 000$

```
1 threads: SERIAL-----  
  
SIMPSON  
Introduce the number of nodes:  
100000  
Energia consumida em Joules: 0.00903  
Approximated pi value: 3.141592653589787  
Total execution time: 0.00138417503330857 sec  
  
MONTECARLO  
Introduce the number of iterations n:  
100000  
Energia consumida em Joules: 0.02289  
Approximate pi value: 3.1348799999999988887111612712  
Final execution time: 0.00194292300147936 sec  
  
MONTECARLOpyhton  
100000  
3.148  
Energia: 0.405638 J  
Potencia: 16.285450457684277  
tempo final: 0.02477264404296875  
  
INTEGRAL  
Introduce the number of iterations n:  
100000  
pi eh 3.141593  
Energia consumida em Joules: 0.01538  
Tempo: 0.001628  
  
8 threads: PARALELO-----  
  
INTEGRAL  
Introduce the number of iterations n:  
100000  
pi eh 3.141593  
Energia consumida em Joules: 0.10693  
Tempo: 0.004070
```


Conclusões: Versão Python e C

```
MONTECARLOpyhton  
100000  
3.13496  
Energia: 0.420897 J  
Potencia: 16.66324874302229  
tempo final: 0.02514934539794922
```

- Outra fato interessante é que para podermos ter o mesmo tempo médio de execução em Python em comparação com C, tivemos que diminuir o n em 10x, ocasionando uma diferença de 0.0063 em média no valor do PI. Mesmo diminuindo o n, ou seja, a quantidade de processamento, o Python gasta mais energia, pelo efeito da modo de compilação, do que feito na linguagem c.

*Foi feito 10 medidas e usado a média de tempo e erro.

Conclusões: Paralelismo

- A Soma com 8 threads é 3x mais rápida que a sequencial.
- A energia gasta decai somente 30% em média.

Conclusões: flags gcc

- Quando trocamos as flags de compilação para gerar objetos e executável menores e com tempo de execução mais rápida, temos:
 - O1 : aplica uma otimização moderada, mas não degrada significativamente o tempo de compilação;
 - O2 : realiza otimização total, gerando o código altamente otimizado e com o tempo mais lento de compilação;

PARA $n = 1\,000\,000$, MAS
USANDO A FLAG -O1

```
gcc -Wall -fopenmp -O1 integral.o energia.o -o pi_integr

1 threads: SERIAL-----

SIMPSON
Introduce the number of nodes:
1000000
Energia consumida em Joules: 0.05676
Approximated pi value: 3.141592653589788
Total execution time: 0.00741462002042681 sec

MONTECARLO
Introduce the number of iterations n:
1000000
Energia consumida em Joules: 0.21075
Approximate pi value: 3.1421600000000006409095476556
Final execution time: 0.0245798449614085 sec

MONTECARLOpyhton
1000000
3.138544
Energia: 4.278127 J
Potencia: 17.417594586782073
tempo final: 0.2453148365020752

INTEGRAL
Introduce the number of iterations n:
1000000
pi eh 3.141593
Energia consumida em Joules: 0.02075
Tempo: 0.001796

8 threads: PARALELO-----

INTEGRAL
Introduce the number of iterations n:
1000000
pi eh 3.141593
Energia consumida em Joules: 0.02679
Tempo: 0.001839
```

$n = 1\,000\,000$, SEM
FLAGS

```
1 threads: SERIAL-----

SIMPSON
Introduce the number of nodes:
1000000
Energia consumida em Joules: 0.13892
Approximated pi value: 3.141592653589788
Total execution time: 0.0177649399847724 sec

MONTECARLO
Introduce the number of iterations n:
1000000
Energia consumida em Joules: 0.19885
Approximate pi value: 3.1415120000000000821671619633
Final execution time: 0.0270153420278803 sec

MONTECARLOpyhton
1000000
3.141288
Energia: 4.278859 J
Potencia: 17.15296669112017
tempo final: 0.24910306930541992

INTEGRAL
Introduce the number of iterations n:
1000000
pi eh 3.141593
Energia consumida em Joules: 0.18439
Tempo: 0.012623

8 threads: PARALELO-----

INTEGRAL
Introduce the number of iterations n:
1000000
pi eh 3.141593
Energia consumida em Joules: 0.11279
Tempo: 0.004264
```

PARA $n = 1\,000\,000$, MAS USANDO A FLAG -O2

```
gcc -Wall -fopenmp -O2 integral.o energia.o -o pi_integral

1 threads: SERIAL-----

SIMPSON
Introduce the number of nodes:
1000000
Energia consumida em Joules: 0.06799
Approximated pi value: 3.141592653589788
Total execution time: 0.0069263480254449 sec

MONTECARLO
Introduce the number of iterations n:
1000000
Energia consumida em Joules: 0.20294
Approximate pi value: 3.1411240000000000271995759249
Final execution time: 0.0222463820246048 sec

MONTECARLOpyhton
1000000
3.141464
Energia: 4.203725 J
Potencia: 17.104513624693308
tempo final: 0.24538874626159668

INTEGRAL
Introduce the number of iterations n:
1000000
pi eh 3.141593
Energia consumida em Joules: 0.03986
Tempo: 0.001770

8 threads: PARALELO-----

INTEGRAL
Introduce the number of iterations n:
1000000
pi eh 3.141593
Energia consumida em Joules: 0.02502
Tempo: 0.001369
```

PARA $n = 1\,000\,000$, SEM FLAGS

```
1 threads: SERIAL-----

SIMPSON
Introduce the number of nodes:
1000000
Energia consumida em Joules: 0.13892
Approximated pi value: 3.141592653589788
Total execution time: 0.0177649399847724 sec

MONTECARLO
Introduce the number of iterations n:
1000000
Energia consumida em Joules: 0.19885
Approximate pi value: 3.14151200000000000821671619633
Final execution time: 0.0270153420278803 sec

MONTECARLOpyhton
1000000
3.141288
Energia: 4.278859 J
Potencia: 17.15296669112017
tempo final: 0.24910306930541992

INTEGRAL
Introduce the number of iterations n:
1000000
pi eh 3.141593
Energia consumida em Joules: 0.18439
Tempo: 0.012623

8 threads: PARALELO-----

INTEGRAL
Introduce the number of iterations n:
1000000
pi eh 3.141593
Energia consumida em Joules: 0.11279
Tempo: 0.004264
```

Fontes

- <http://recologia.com.br/2013/07/25/estimando-o-valor-de-pi-usando-o-metodo-de-monte-carlo/>
- https://pt.wikipedia.org/wiki/F%C3%B3rmula_de_Simpson
- https://en.wikipedia.org/wiki/Riemann_sum
- Códigos em <https://github.com/LeDSantos/Pi-calculation>