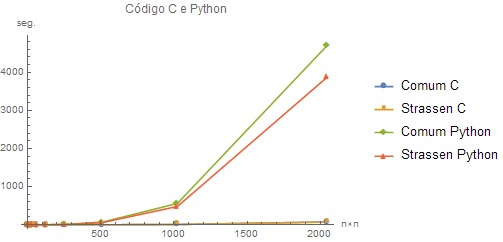
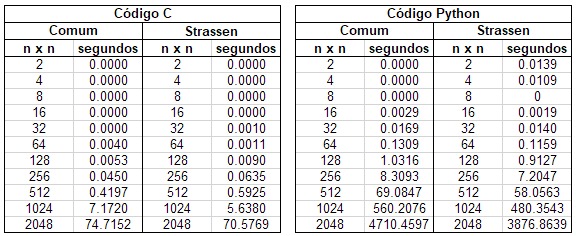
**COMPARAÇÃO ENTRE A PERFORMANCE DOS ALGORITMOS**

Metodologia: Para obter-se uma comparação entre o desempenho os algoritmos desenvolvidos, executou-se cada algoritmo uma vez para matrizes de diferentes ordens. As ordens escolhidas foram sempre potências de 2, iniciando em 2 e terminando em 2048. Com a finalidade de obter-se uma comparação entre a variação entre diferentes linguagens, criou-se códigos equivalentes em Python e em C. Para manter a coerência das medidas, executou-se os códigos em um mesmo computador, e sem outros aplicativos abertos no momento do teste.

Resultados: A tabela e os gráficos comparativos abaixo representam os resultados obtidos a partir da execução dos códigos elaborados em Python e em C. A métrica utilizada para a comparação foi o tempo, medido em segundos utilizando-se tomadas de tempo no início e no final do cálculo, com as funções *datetime.datetime.now()*, em Python, e gettimeofday(), em C.



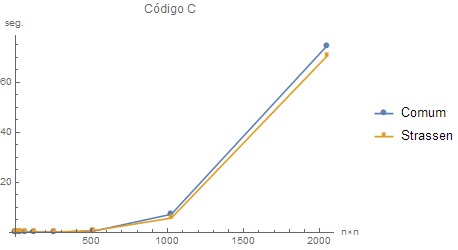
**Gráfico 1:** Comparação dos resultados obtidos



**Tabela 1:** Rendimento dos algoritmos para diferentes tamanhos de matrizes

Discussão: À primeira vista, é bastante destacada a diferença em rendimento entre os algoritmos equivalentes em Python e em C, o que nos permite concluir que os códigos desenvolvidos em Python foram menos eficientes que os desenvolvidos em C.

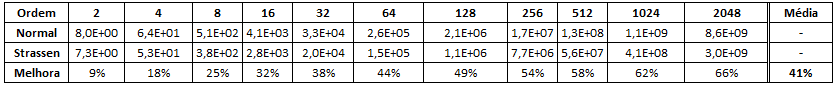
No código em C, observou-se uma melhora de aproximadamente 6% no desempenho para matrizes de ordem 2048, e de aproximadamente 2% pra matrizes de ordem 1024. Por outro lado, observou-se que para matrizes menores, o algoritmo de Strassen apresenta rendimento inferior, chegando a ser quase 70% mais lento no caso de matrizes de ordem 128.



**Gráfico 2:** Rendimentos dos algoritmos de multiplicação desenvolvidos em C

Já no código em Python, o algoritmo de Strassen apresenta rendimento superior para todas as matrizes com ordem maior ou igual a 16, tendo uma média geral de cerca de 17% de melhora no desempenho em relação ao algoritmo convencional.

Com essas informações em mente, percebe-se que quanto mais computacionalmente custoso o código, maior é a melhora obtida pela substituição do algoritmo tradicional pelo algoritmo de Strassen nas operações de multiplicação de matrizes.

or fim, percebe-se que mesmo alcançando uma melhora considerável utilizando o código em Python, a melhora não alcançou a melhora teórica média, que seria de aproximadamente 41% (utilizando-se n³ para o algoritmo normal e n2,86 para o de Strassen) para o mesmo intervalo de testes. Isso se deve principalmente ao fato de que o cálculo realizado leva em consideração somente as operações de multiplicação, descartando as somas, subtrações, atribuições, etc.

**Tabela 2:** Análise da melhora teórica esperada pela substituição