# Strassen Algorithm Implementation

Maria Grazia Berni

Results and Brief description of Implementations

Observing recursive calls of Strassen algorithm, it was found that it is necessary to allocate only one initial matrix S at the beginning of the procedure, and having the same size of the input matrices. Using this matrix as a support matrix, where we save partial results, and also using the final matrix where the product is saved, to save partial and final results, a lot of memory is saved. Here a plot of the execution time of the naive implementation of the Strassen algorithm, and of the optimized version of it.
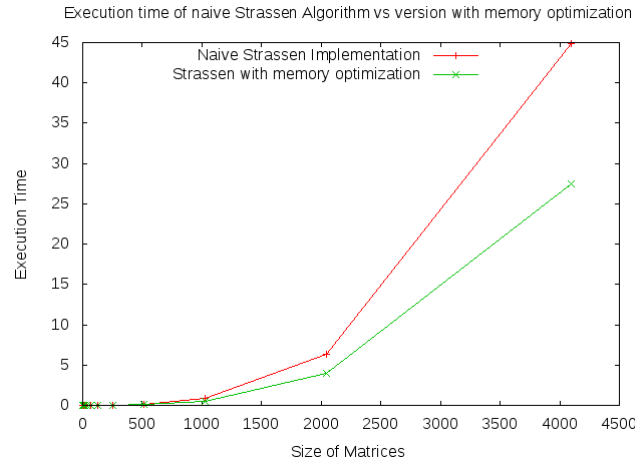
To deal with rectangular matrices, in the matrix multiplication the two matrices, with dimensions $(a * m)$ and $(m * b)$, have been resized considering the largest size among a, m and b, and increasing its value until it reaches a power of two number. In order to do this, the excess parts of the matrices were filled using zeros. It follows a plot of the execution time of the generalized Strassen Algorithm and the execution time of the naive rectangular matrix multiplication. The naive algorithm, of course, deal with the original sizes of the two matrix, so here n refers to the square filled matrices.

Execution time generalized Strassen Algorithm vs naive Algorithm
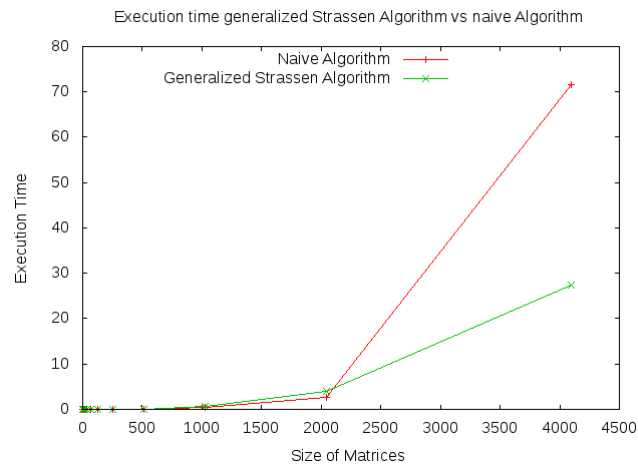
Only for very large sizes of the input matrices, the generalized Strassen algorithm becomes faster then the naive rectangular matrix multiplication, while, for square matrices, it becomes faster starting from smaller input sizes, as shown below.

Execution time of optimized Strassen Algorithm vs naive Algorithm