

Solution of serie 5

Exercise 1 *The result depend of course on the machine, but the qualitative behavior should be the same. For both compilers, one notices an important effect of the operations prefetch and unroll; the operation unroll+prefetch is quite efficient in both cases. For the standard loop and the different optimization options, the influence of loop unrolling is small. For both compilers, the best option is O3, but it is still worse than the hand-coded version. The automatic vectorization is rather disappointing, especially for the Intel compiler.*

Exercise 2 *One notices the version by rows is clearly better, because the two dimensional arrays are stored by rows in C. This remains true even without vectorization (icc loop1.c, gcc44 loop1.c). Furthermore, the Intel compiler with the options -O3 -march=core2 is able to permute the column version.*

Exercise 3 *Their is almost no benefit of using the Blas in this case, compared to the vectorized or non vectorized versions. It is also true for the optimized Blas. In both cases, using the dot product is better. For the Mkl library and the Intel compiler, one has about the same results. .*

Exercise 4 *The speedup is given by*

$$S(n) = \frac{1}{f_p/n + f_s} = \frac{1}{1 + f_p(1/n - 1)}.$$

The plots are obtained by the Matlab script `speed.m`. Despite its rather optimistic nature, the Amdahl's law shows that one needs the parallelization of almost the whole code if one wants a good speedup on many processors.