

Exercise 3

SDS 335

Carly Smith

icc compiler optimization run times:

-O0: 3.918440e-01

-O1: 3.785911e-01

-O2: 4.053116e-06

-O3: 2.861023e-06

gcc would not compile the code – it said the reference to the sin and cos functions were undefined, even with `#include <math.h>` at the top of the file.

The compilation optimizer likely uses **constant propagation** to reduce the runtime on the main function. Doubles x and y are always .5 and alpha is always 1.57 in this function, so the rotate function could be called with (.5,.5, 1.57) to reduce time calling the variables.

Constant Folding is likely not used, since there is no simple arithmetic among constants that could be simplified. Additionally, **Algebraic Simplification** is probably not used since there aren't any instances of an operation with identities. Many of the other optimization transformations don't apply to the rotate.c file since the code doesn't have any recursion, dead code, duplicate subexpressions, etc. It is my guess that since the time for -O0 and -O1 as well as -O2 and -O3 was so close to each other that the majority of constant propagation happened between levels 1 and 2. The other optimizations would have taken place between O0 and O1/O2 and O3 which didn't affect the code runtime very much. Since O3 is known for being dangerous, the risky optimizations like common subexpression elimination and probably take place here.