

Title:  
Matrix Multiplication and Quantum Monte Carlo (QMC)  
on Graphical Processing Units (GPUs)

Authors\*:  
Amos Anderson[1]  
William A Goddard, III  
Peter Schroder

\* California Institute of Technology  
1200 E California Blvd  
Pasadena, CA 91125

[1] [amosa@caltech.edu](mailto:amosa@caltech.edu)

Abstract:

GPUs have been widening their multimedia acceleration applicability to include general purpose computations in recent years. With the addition of enhanced functionality on the hardware and improved software interface, it has become a processor worth considering for scientific calculations. Although the GPU is not necessarily suitable for arbitrary types of computations, the range of candidates is growing. Our QMC software has been studied for compatibility with the GPU. This involves matrix multiplication as well as a couple of quantum chemistry specific kernels.

We report here performance results and rounding error correction for matrix multiplication on 7800 GTX nVidia GPU. We show that the GPU can compete quite well against the ATLAS BLAS library on a Pentium 4 3.0 Ghz for dense matrix multiplication in a wide variety of matrix dimensions. For many applications, achieving a break even with a CPU can be useful since it would permit a continuous flow of computational stages on the same processor.

Due to differences between how a GPU handles floating point numbers and the IEEE floating point specification, each addition and multiplication operation can introduce small rounding errors. When left uncorrected, the propagated rounding error from each inner product scales approximately linearly with length. A solution to this problem was presented by Kahan in 1965 which can recover the rounding error accumulated when summing a sequence of numbers. With slight GPU specific modifications this technique, the Kahan Summation Formula (KSF), has been applied resulting in relative errors that are approximately constant with respect to matrix dimensions.

Although matrix multiplication on the GPU has already been studied by several people, we are interested in some specific “real world” optimizations, especially for the speedup gains possible for smaller matrices and for rectangular matrices. Furthermore, we also wanted to explore the speedup gains where the multiplicand was reused. For smaller matrices, the memory read cost quenches GPU performance gains relative to CPU benchmarks. For large enough matrices a saturation point is reached in the number of

floating point operations per second (flops). The challenge is thus to make this saturation point available to as small of matrices as possible.

The results have shown that using a variety of tricks and tuning procedures, we are able to obtain satisfactory speedups in nearly all cases. In some domains where CPU cache peculiarities play a role, the GPU is occasionally able to significantly outperform the CPU. When the KSF option is applied, the technique usually puts the speed on par with the CPU. Both the reuse of multiplicands and the use of rectangular matrices (with lengthened inner products) are able to bring the saturation point down significantly.

Lastly, a few QMC specific kernels were streamlined and ported to the GPU. Phenomenal speedups were achieved relative to CPU performance, albeit curbed by an Amdahl's Law type of diminishing returns. Errors relating to a lack of denormals was investigated, but it was observed that these errors were small relative to the errors in the linear algebra.