JCTC Journal of Chemical Theory and Computation

ARTICLE

pubs.acs.org/JCTC

Dynamic Precision for Electron Repulsion Integral Evaluation on Graphical Processing Units (GPUs)

Nathan Luehr, Ivan S. Ufimtsev, and Todd J. Martínez*

PULSE Institute and Department of Chemistry, Stanford University, Stanford, California 94305, United States

SLAC National Accelerator Laboratory, Menlo Park, California 94025, United States

Supporting Information

ABSTRACT: It has recently been demonstrated that novel streaming architectures found in consumer video gaming hardware such as graphical processing units (GPUs) are well-suited to a broad range of computations including electronic structure theory (quantum chemistry). Although recent GPUs have developed robust support for double precision arithmetic, they continue to provide 2−8× more hardware units for single precision. In order to maximize performance on GPU architectures, we present a technique of dynamically selecting double or single precision evaluation for electron repulsion integrals (ERIs) in Hartree−Fock and density functional self-consistent field (SCF) calculations. We show that precision error can be effectively controlled by evaluating only the largest integrals in double precision. By dynamically scaling the precision cutoff over the course of the SCF procedure, we arrive at a scheme that minimizes the number of double precision integral evaluations for any desired accuracy. This dynamic precision scheme is shown to be effective for an array of molecules ranging in size from 20 to nearly 2000 atoms.

INTRODUCTION

It has recently been recognized that consumer video game hardware is well suited to many tasks in computational chemistry, including electronic structure theory,[1−10] *ab initio* molecular dynamics,[11] and empirical force-field-based molecular dynamics.[8,12−14] The emergence of the CUDA development framework from NVIDIA has made it much easier to repurpose this hardware for scientific computing,[15] compared to early efforts on similar architectures that had to resort to low level instructions.[16] Nevertheless, efficient use of graphical processing units (GPUs) requires careful attention to some specialized hardware constraints such as memory access patterns and nonuniform efficiency of floating point arithmetic in different precision. Furthermore, GPUs have been carefully designed for maximum performance in specific graphics processing tasks and are otherwise severely limited. It is unlikely that these limitations will be fully eliminated because in large part they provide the foundation of the GPUs computational prowess.

The first CUDA-enabled GPUs had no support for double precision arithmetic, demanding care in their use for quantum chemistry applications. The latest GPUs fully support double precision arithmetic, with stunning performance in the range of several hundred GFLOPS, well beyond that of traditional processors (CPUs). Nevertheless, single precision continues to maintain between 2× and 8× more instruction units than double precision on the latest generation of GPUs. This disparity stems from the hardware's pedigree in graphics, where there is little need for double precision accuracy, and the necessary increase in circuitry is difficult to justify. Single precision may exhibit further performance advantages as a result of its smaller memory footprint, which reduces data bandwidth requirements[2] and increases the number of values that can be cached in registers. Thus, for maximum performance, it remains important to favor single precision arithmetic as much as possible on GPUs.

To balance GPU performance with chemical accuracy, quantum chemistry implementations have adopted mixed precision approaches in which double precision operations are added sparingly to an otherwise single precision calculation. Matrix multiplication in the context of resolution-of-the-identity Møller−Plesset perturbation theory has been shown to provide accurate mixed precision results, even when the majority of operations are carried out in single precision.[4,6] Single precision ERI evaluation has been successfully augmented with double precision accumulation into the matrix elements of the Coulomb and exchange operators.[3,5] "Double precision accumulation" simply means that the ERIs are evaluated in single precision, but a double precision variable is used to accumulate the products of density matrix elements and ERIs which make up the final operator (e.g., Coulomb or exchange). For example, the Coulomb operator can be constructed as

$$J^{64}_{\mu\nu} \mathrel{+}= P^{32}_{\lambda\sigma}(\mu\nu|\lambda\sigma)^{32} \tag{1}$$

where the superscripts indicate the number of bits of precision used for the labeled variable, and the ERIs are given as

$$(\mu\nu|\lambda\sigma) = \int \frac{\phi_\mu(r_1)\,\phi_\nu(r_1)\,\phi_\lambda(r_2)\,\phi_\sigma(r_2)}{|r_1 - r_2|} \mathrm{d}r_1\,\mathrm{d}r_2 \tag{2}$$

Computing a few of the largest ERIs in full double precision has also been shown[5] to improve accuracy compared to calculations using only single precision for all ERIs. Incremental construction of the Fock matrix[17] has been noted to improve the accuracy of

Received: December 6, 2010
Published: March 23, 2011

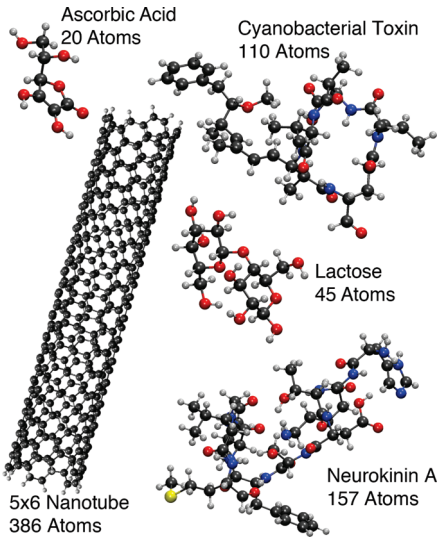

Figure 1. Molecular geometries used to benchmark the correlation between precision cutoff and the effective precision of the final energy. Optimized geometries (shown here) were used in addition to distorted nonequilibrium geometries prepared by carrying out RHF/STO-3G NVT dynamics at 2000 K (1000 K for ascorbic acid).

single precision ERI evaluation.[3,4] Finally, it has been suggested that full single precision can be safely employed in the earliest SCF iterations.[5] Such strategies have proven effective in improving mixed precision results for many calculations reported to date. However, no systematic study of mixed precision ERI calculation has been undertaken. In fact, the precision cutoffs must be chosen carefully for each molecular system studied to guarantee that the absolute error is within a tolerable range.

In the present paper, we introduce a systematic method of both controlling error and minimizing double precision operations in mixed precision Fock matrix construction. We begin by describing a mixed precision scheme, in which all integrals are computed on the GPU, with large ERIs calculated in full double precision and small integrals in single precision with double precision accumulation. We show that the relative error in such calculations is well behaved for an array of systems and can be controlled to provide an effective precision and performance between that of single and full double precision. In order to further decrease the number of double precision integral evaluations, we suggest a new method that we term *dynamic precision*, in which the effective precision is adjusted dynamically between SCF iterations. In this way, the minimum number of double precision integrals can be used to obtain any desired level of accuracy. Finally, we present performance and accuracy results to benchmark the dynamic precision approach. The mixed and dynamic precision schemes were implemented in TeraChem, a general purpose quantum chemistry package designed specifically for the GPU.[11,18]

MIXED PRECISION IMPLEMENTATION

The magnitude of an ERI is commonly bounded using the Schwarz inequality.[19] In direct SCF codes,[17] the bound can be further reduced using elements of the density matrix as

$$|(\mu\nu|\lambda\sigma)P_{\lambda\sigma}| \leq (\mu\nu|\mu\nu)^{1/2}(\lambda\sigma|\lambda\sigma)^{1/2}|P_{\lambda\sigma}| \qquad (3)$$

Because only absolute accuracy is required in chemical applications, Yasuda introduced a cutoff on the density-weighted Schwarz bound to group ERIs into two batches. Those whose bound fell below the cutoff were calculated in single precision on the GPU. Integrals whose bound was larger than the cutoff were evaluated in double precision on the CPU. As has been previously noted,[1] the use of the CPU is no longer necessary with the advent of robust double precision support on the GPU. We have implemented a mixed precision Fock matrix evaluation scheme similar to that suggested by Yasuda. Instead of using the CPU, however, we developed double precision analogues of our previously reported single precision ERI routines. Implementation details follow those in our previously described single precision code.[3]

In order to make the most of the GPU's memory bandwidth, the double and single precision integrals are handled in a two-pass algorithm. As previously described, our ERI algorithm operates directly on primitive Gaussian pair products, which are sorted by decreasing Schwarz bound. In the first pass, data for the largest primitive pairs are packed into double precision arrays, and any ERI whose bound is greater than the precision threshold is calculated using double precision GPU kernels. In the second pass, smaller primitive pairs are added to the data, which is reassembled into single precision arrays and processed by single precision kernels.

Because the four-index, density-weighted Schwarz bound is computed only within the GPU kernels, some duplication occurs between the sets of single and double precision primitive pairs, and each kernel must filter out individual ERIs whose bound falls outside of the relevant range. When filtering, it is essential that both the single and double precision kernels handle the bounds identically. Otherwise, the different rounding behavior exhibited by single and double precision arithmetic will cause integrals close to the bound to be skipped or double counted. In our implementation, the double precision kernels cast the bound quantities to single precision before determining whether the associated integrals and their contributions will be evaluated.

Table 1. RHF/6-31G Final Energies Compared between GAMESS (Set at Default Convergence and Two-Electron Integral Thresholds), Our GPU Accelerated TeraChem Code Performing All Calculations in Double Precision (TeraChem DP), and TeraChem Using Single Precision for ERIs with Double Precision Accumulation into the Fock Matrix Elements (TeraChem SP)[a]

	ascorbic acid	lactose	cyanobacterial toxin
GAMESS	−680.5828947	−1289.6666250	−2491.2058893
TeraChem DP	−680.5828947	−1289.6666250	−2491.2058890
TeraChem SP	−680.5828071	−1289.6664266	−2491.2053589
	neurokinin A		5 × 6 nanotube
GAMESS	−4089.6883770		−13790.1415171
TeraChem DP	−4089.6883762		−13790.1415176
TeraChem SP	−4089.6879824		−13790.1389987

[a] Distorted nonequilibrium geometries from RHF/STO-3G NVT dynamics at 2000 K (1000 K for ascorbic acid) were used.

MIXED PRECISION ACCURACY

The molecules shown in Figure 1 were chosen as representative test cases to study the accuracies of several mixed precision

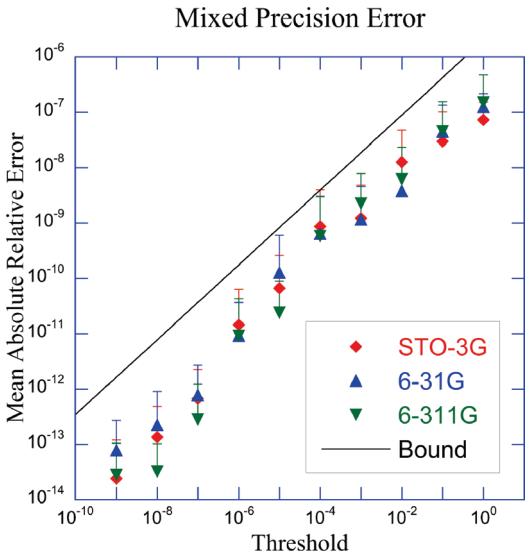

Figure 2. Average relative error in final RHF energies versus the precision threshold. Both minimized and distorted nonequilibrium geometries for the molecules in Figure 1 are included in averages. Error bars represent two standard deviations above the mean. The black line represents the empirical error bound given in eq 4.

thresholds. Geometries were prepared at both an optimized RHF/6-31G minimum and a distorted geometry obtained by performing NVT dynamics at ~2000 K. Coordinates are provided in Table S1 of the Supporting Information. The double precision GPU implementation was first benchmarked against GAMESS[20] using GAMESS default convergence and two-electron integral thresholds. The resulting final energies, shown in Table 1, indicate good agreement. Switching to single precision ERIs degrades the result by 3−5 orders of magnitude. However, it should be emphasized that for molecules containing as many as 100 atoms, single precision provides adequate results—the absolute error of the energy computed with single precision ERIs is well below 1 kcal/mol even for Neurokinin A with 157 atoms.

Next, we evaluated the mixed precision approach by varying the precision threshold between 10^{-9} au (nearly all integrals are evaluated in double precision) and 1.0 au (essentially all integrals use single precision). Negligibly small ERIs were screened according to the density weighted Schwarz upper bound of eq 3 using a conservative threshold to ensure that differences in the final energy above $\sim 10^{-7}$ au would be dominated by mixed precision errors. The average relative energy difference between full double and mixed precision SCF energies for the 10 test geometries described above is plotted as a function of the precision threshold in Figure 2. Although the absolute error increases along with the total energy of the systems, the relationship between the precision cutoff and relative error is roughly linear and is independent of system size and basis set. Thus, each threshold can be associated with an effective relative error in the energy.

When a power fit of the average errors in Figure 2 is used, it is possible to empirically preselect a precision threshold corresponding to any accuracy requirement using only the estimated total energy (to calculate relative error). A reasonably conservative bound on the precision error was obtained by shifting a power fit of the average 6-31G errors beyond two standard

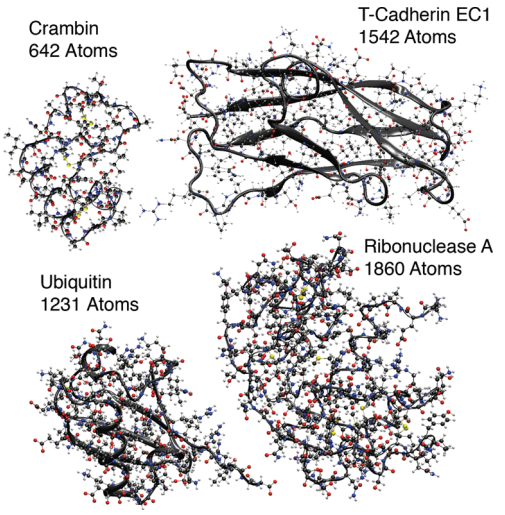

Figure 3. Additional molecules used to test the dynamic precision algorithm.

deviations above the mean. The result is given in eq 4 and plotted for reference in Figure 2.

$$\mathrm{Err}(\mathrm{Thre}) = 2.0 \times 10^{-6}\mathrm{Thre}^{0.7} \quad (4)$$

By inverting eq 4, we can select an adequate precision threshold at the start of the SCF procedure and use the minimum allowable effective precision. However, this requires early iterations whose density matrices are highly approximate to use the full level of precision needed at convergence. This is especially wasteful in very large systems since the accuracy required at convergence nears that of full double precision. To further reduce the use of double precision in early iterations while still achieving the required accuracy at convergence, we introduce a dynamic precision approach, described below.

DYNAMIC PRECISION IMPLEMENTATION

The essence of the dynamic precision approach is to use eq 4 to select a different threshold for each iteration of the SCF procedure. Early iterations have been shown to tolerate relatively large errors in the Fock matrix without hampering convergence.[3,21] We take the maximum element of the DIIS error vector[22] from the previous iteration as a metric of this tolerance, and at each iteration, we use eq 4 to select a threshold providing precision safely below the DIIS error. This ensures that the precision error is a small contributor to the total error. By reducing the precision threshold gradually as convergence progresses, it is possible to approach full double precision results while minimizing the number of actual double precision operations.

To improve performance, our SCF code uses an iterative update approach (also known as incremental Fock matrix formation) to build up the Fock matrix over the course of the SCF procedure.[17] The update approach decomposes the Fock matrix as

$$F_{i+1}(P_{i+1}) = F_i(P_i) + F(P_{i+1} - P_i) \quad (5)$$

so that only the last term needs to be calculated in each SCF iteration. Here P_i and $F_i(P_i)$ are the density and Fock matrices at the ith SCF iteration. Because changes in the density matrix become very small near convergence, the iterative Fock approach

Table 2. Comparison of Double and Dynamic Precision Final RHF/6-31G Energies (Listed in Hartree)[a]

	double precision		dynamic precision			
	final energy	iter	final energy	iter	precision error	conv thres
ascorbic acid (minimum)	−680.6986413210	16	−680.6986413153	16	5.70×10^{-9}	10^{-7}
	−680.6986413213	12	−680.6986413898	12	6.85×10^{-8}	10^{-5}
ascorbic acid (1000 K)	−680.5828947151	17	−680.5828947066	17	8.50×10^{-9}	10^{-7}
	−680.5828947060	12	−680.5828947665	12	6.05×10^{-8}	10^{-5}
lactose (minimum)	−1290.0883460632	14	−1290.0883460414	14	2.18×10^{-8}	10^{-7}
	−1290.0883460086	10	−1290.0883459660	10	4.26×10^{-8}	10^{-5}
lactose (2000 K)	−1289.6666249592	15	−1289.6666249614	15	2.20×10^{-9}	10^{-7}
	−1289.6666249365	11	−1289.6666248603	11	7.62×10^{-8}	10^{-5}
cyano toxin (minimum)	−2492.3971992758	19	−2492.3971992873	19	1.15×10^{-8}	10^{-7}
	−2492.3971992730	13	−2492.3971985116	13	7.61×10^{-7}	10^{-5}
cyano toxin (2000 K)	−2491.2058890235	21	−2491.2058890017	21	2.18×10^{-8}	10^{-7}
	−2491.2058889916	13	−2491.2058886707	13	3.21×10^{-7}	10^{-5}
neurokinin A (minimum)	−4091.3672645555	19	−4091.3672645944	20	3.89×10^{-8}	10^{-7}
	−4091.3672645489	14	−4091.3672644494	14	9.95×10^{-8}	10^{-5}
neurokinin A (2000 K)	−4089.6883762179	21	−4089.6883761946	21	2.33×10^{-8}	10^{-7}
	−4089.6883760772	15	−4089.6883758130	15	2.64×10^{-7}	10^{-5}
nanotube (minimum)	−13793.7293925221	24	−13793.7293925323	23	1.02×10^{-8}	10^{-7}
	−13793.7293924922	15	−13793.7293928287	15	3.37×10^{-7}	10^{-5}
nanotube (2000 K)	−13790.1415175662	29	−13790.1415175584	27	7.80×10^{-9}	10^{-7}
	−13790.1415175332	18	−13790.1415191026	18	1.57×10^{-6}	10^{-5}
crambin	−17996.6562925538	18	−17996.6562926036	18	4.98×10^{-8}	10^{-7}
	−17996.6562925535	12	−17996.6562927894	12	2.36×10^{-7}	10^{-5}
ubiquitin	−29616.4426376594	24	−29616.4426376596	24	2.00×10^{-10}	10^{-7}
	−29616.4426376302	18	−29616.4426376655	18	3.53×10^{-8}	10^{-5}
T-cadherin EC1	−36975.6726049407	21	−36975.6726049394	21	1.30×10^{-9}	10^{-7}
	−36975.6726049265	16	−36975.6726048777	16	4.88×10^{-8}	10^{-5}
ribonuclease A	−50813.1471248227	19	−50813.1471248179	19	4.80×10^{-9}	10^{-7}
	−50813.1471247051	12	−50813.1471250247	12	3.20×10^{-7}	10^{-5}

[a] Precision error is taken as the absolute difference between double and dynamic precision energies. The number of SCF iterations required to reach convergence is listed (iter) as well as the threshold used to converge the maximum element of the DIIS error matrix.

allows many additional integrals to be screened. Typically, this provides an overall speedup between 2× and 3× over the conventional SCF approach. However, the naïve implementation of dynamic precision described above causes the iterative Fock method to converge incorrectly, because each update of the Fock matrix does not compensate for the precision error of the previous step.

Rather than abandoning the iterative Fock algorithm altogether, we introduce the following adjustment. When the relative DIIS error drops below the error bound of the current precision threshold, the threshold is reduced to provide enough accuracy for a several orders of magnitude reduction in the DIIS error. Each time the precision is improved, the Fock matrix is recalculated from scratch. Between threshold reductions, the faster iterative Fock update scheme can be safely employed.

RESULTS

To benchmark our dynamic precision approach, we performed RHF energy calculations on the test geometries presented above, as well as some larger systems shown in Figure 3. Table 2 demonstrates the accuracy provided by our dynamic precision approach. In each calculation, the dynamic precision method is successful in reproducing the full double precision results to within the convergence criteria. Furthermore, the number of SCF iterations required to reach convergence (also shown in Table 2) is essentially identical between dynamic and double precision. Finally, the SCF energy difference between dynamic and double precision remains fairly constant over the range of test systems, indicating that our empirical error bound is reasonably calibrated.

Of course, as with full double precision, the energy is expected to converge more rapidly than properties with first-order wave function dependence, and the final precision threshold may need to be stricter than required by eq 4. This is exactly analogous to the stricter convergence criteria that are routinely used in many quantum chemistry packages when treating first-order properties. It should be noted that in the present scheme tightening the convergence criteria will automatically reduce the final precision bound. However, a detailed analysis of precision cutoffs suitable for gradient and other property calculations is left as a topic of future research.

Table 3 summarizes the performance of our algorithm on two GPU platforms. On the older Tesla C1060 GPU, dynamic precision accelerates the SCF calculation by up to 4× over full double precision. The Tesla C2050 includes a greater proportion of double precision units, and as a result the performance margin

Table 3. Runtime Comparison between Double and Dynamic Precision for RHF/6-31G Single Point Energy Calculations Converged to a Maximum DIIS error of 10^{-5} au[a]

	Nvidia Tesla C1060			Nvidia Tesla C2050		
	double runtime	dynamic runtime	speedup	double runtime	dynamic runtime	speedup
ascorbic acid	7.65	2.23	3.4	4.43	2.93	1.5
lactose	36.82	9.70	3.8	15.79	8.41	1.9
cyano toxin	352.74	87.66	4.0	156.79	68.44	2.3
neurokinin A	734.61	197.91	3.7	337.68	149.76	2.3
nanotube	4693.99	1716.88	2.7	3042.92	1155.58	2.6
crambin	2754.35	1104.22	2.5	1390.49	762.09	1.8
ubiquitin	29674.48	11833.58	2.5	14997.94	7517.68	2.0
T-cadherin EC1	40936.81	17408.21	2.4	20889.98	10781.42	1.9
ribonuclease A	50092.20	21869.37	2.3			

[a] Times are given in seconds and represent the wall time of the entire calculation. The hardware platform included dual Intel Xeon X5570 CPUs, 72 gigabytes of RAM, and eight GPUs. Only one GPU was used for the smaller systems (ascorbic acid through neurokinin A) to ensure that it remained saturated with parallel work. The Tesla C2050 could not treat ribonuclease A at the RHF/6-31G level due to memory constraints.

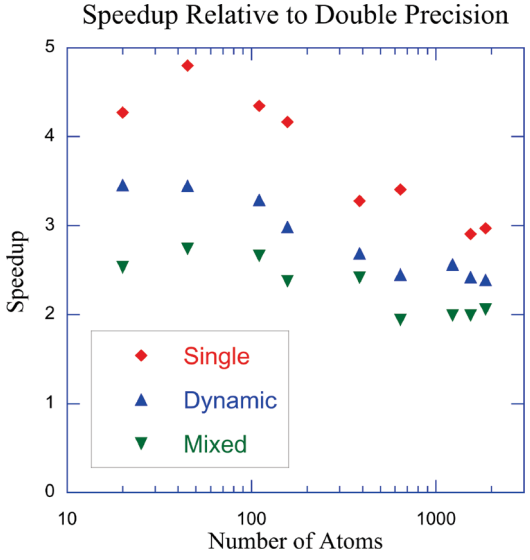

Figure 4. Speedups for RHF single point energy calculations using single, dynamic, and mixed precision relative to full double precision performance. Calculations were run on an Nvidia Tesla C1060 GPU and were converged to a DIIS error of 10^{-7} au. Mixed precision calculations used a static precision threshold chosen for each system by inverting eq 4 and solving for an absolute accuracy of 10^{-7} Hartree. Single precision failed to converge for ubiquitin.

between double and single precision arithmetic is narrowed. However, even here, dynamic precision accelerates the overall energy calculation by a full 2×. Figure 4 compares the wall clock speedups provided by dynamic, mixed, and single precision over full double precision calculations on the Tesla C1060 GPU. Here, *mixed precision* refers to statically fixing the precision threshold for the entire SCF procedure at the value prescribed by solving eq 4 for an error of 10^{-7} Hartrees. Dynamic precision consistently outperforms the simpler mixed precision scheme despite requiring periodic rebuilds of the Fock matrix. More importantly, dynamic precision consistently provides between 70 and 80% of the performance of single precision while providing results comparable to full double precision, and this pattern remains intact even for the largest systems.

In Figure 4, the margin between single and double precision decreases for the largest systems. This is most likely the result of the GPU's finite texture cache. In exchange kernels, density matrix elements are accessed out of order, and the texture cache is used to ameliorate noncoalesced memory loads.[3] However, as the system size grows, neighboring threads begin accessing disparate parts of the density matrix and must be serviced by multiple texture loads. In the limit of large systems, the noncoalesced memory access will cause the exchange kernels to be completely memory bound, and single precision should tend to a limit 2× faster than double precision owing to its smaller memory footprint.

CONCLUSION

We have demonstrated that by dynamically adjusting the ratio of integrals calculated in single and double precision on the GPU it is possible to minimize the number of double precision arithmetic operations in constructing the Fock matrix while still systematically controlling the error. Exploiting this flexibility, we have customized our Fock matrix routines for maximum performance on the GPU. Our dynamic precision implementation is able to achieve in excess of 70% of single precision's performance while maintaining accuracy comparable to full double precision. Finally, we have shown that dynamic precision is applicable to systems of unprecedented size.

The same approach can also be adapted for other hardware architectures. On traditional CPUs, for example, the use of single precision arithmetic may improve performance by reducing memory bandwidth or by enabling use of optimized SSE instructions. For extremely large systems, the required relative accuracy may well extend beyond the capacity of double precision.[23] In this limit, the approach outlined above will again prove useful in systematically improving double precision with a minimum of higher precision arithmetic operations. A more comprehensive multiprecision strategy can be easily envisioned, for example, using single, double, and quadruple precision evaluation of different ERIs, according to their magnitude. Furthermore, the same dynamical precision approach can be applied to the calculation of the Coulomb and exchange operators in density functional theory, and similar performance gains will be obtained.

■ ASSOCIATED CONTENT

Supporting Information. Coordinates for all test molecular geometries. This information is available free of charge via the Internet at http://pubs.acs.org/.

■ AUTHOR INFORMATION

Corresponding Author
*E-mail: todd.martinez@stanford.edu.

■ ACKNOWLEDGMENT

This work was supported by NSF (CHE-06-26354) and the Department of Defense (Office of the Director of Defense Research and Engineering) through a National Security Science and Engineering Faculty Fellowship. I.S.U. is an NVIDIA Fellow. Partial support has been provided by PetaChem, LLC through an AFOSR STTR grant administered by Spectral Sciences. T.J.M. and I.S.U. are part owners of PetaChem, LLC.

■ REFERENCES

(1) Ufimtsev, I. S.; Martinez, T. J. *J. Chem. Theory Comput.* **2008**, *4*, 222.
(2) Ufimtsev, I. S.; Martinez, T. J. *Comp. Sci. Eng.* **2008**, *10*, 26.
(3) Ufimtsev, I. S.; Martinez, T. J. *J. Chem. Theory Comput.* **2009**, *5*, 1004.
(4) Vogt, L.; Olivares-Amaya, R.; Kermes, S.; Shao, Y.; Amador-Bedolla, C.; Aspuru-Guzik, A. *J. Phys. Chem. A* **2008**, *112*, 2049.
(5) Yasuda, K. *J. Comput. Chem.* **2008**, *29*, 334.
(6) Olivares-Amaya, R.; Watson, M. A.; Edgar, R. G.; Vogt, L.; Shao, Y. H.; Aspuru-Guzik, A. *J. Chem. Theory Comput.* **2010**, *6*, 135.
(7) Asadchev, A.; Allada, V.; Felder, J.; Bode, B. M.; Gordon, M. S.; Windus, T. L. *J. Chem. Theory Comput.* **2010**, *6*, 696.
(8) Anderson, J. A.; Lorenz, C. D.; Travesset, A. *J. Comput. Phys.* **2008**, *227*, 5342.
(9) Genovese, L.; Ospici, M.; Deutsch, T.; Mehaut, J.-F.; Neelov, A.; Goedecker, S. *J. Chem. Phys.* **2009**, *131*, 034103.
(10) Yasuda, K. *J. Chem. Theory Comput.* **2008**, *4*, 1230.
(11) Ufimtsev, I. S.; Martinez, T. J. *J. Chem. Theory Comput.* **2009**, *5*, 2619.
(12) Friedrichs, M. S.; Eastman, P.; Vaidyanathan, V.; Houston, M.; Legrand, S.; Beberg, A. L.; Ensign, D. L.; Bruns, C. M.; Pande, V. S. *J. Comput. Chem.* **2009**, *30*, 864.
(13) Harvey, M. J.; Giupponi, G.; DeFabiritiis, G. *J. Chem. Theory Comput.* **2009**, *5*, 1632.
(14) Stone, J. E.; Phillips, J. C.; Freddolino, P. L.; Hardy, D. J.; Trabuco, L. G.; Schulten, K. *J. Comput. Chem.* **2007**, *28*, 2618.
(15) Kirk, D. B.; Hwu, W. W. *Programming Massively Parallel Processors: A Hands-On Approach*; Morgan Kauffman: Burlington, MA, 2010; p 1.
(16) Levine, B.; Martinez, T. J. *Abstr. Pap.—Am. Chem. Soc.* **2003**, *226*, U426.
(17) Almlof, J.; Faegri, K.; Korsell, K. *J. Comput. Chem.* **1982**, *3*, 385.
(18) PetaChem. http://www.petachem.com (accessed Feb 1, 2011).
(19) Whitten, J. L. *J. Chem. Phys.* **1973**, *58*, 4496.
(20) Schmidt, M. W.; Baldridge, K. K.; Boatz, J. A.; Elbert, S. T.; Gordon, M. S.; Jensen, J. H.; Koseki, S.; Matsunaga, N.; Nguyen, K. A.; Su, S. J.; Windus, T. L.; Dupuis, M.; Montgomery, J. A. *J. Comput. Chem.* **1993**, *14*, 1347.
(21) Rudberg, E.; Rubensson, E. H.; Salek, P. *J. Chem. Theory Comput.* **2009**, *5*, 80.
(22) Pulay, P. *J. Comput. Chem.* **1982**, *3*, 556.
(23) Takashima, H.; Kitamura, K.; Tanabe, K.; Nagashima, U. *J. Comput. Chem.* **1999**, *20*, 443.