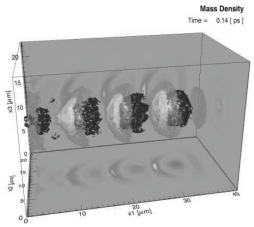
Examining Precision in OSIRIS w/ CUDA

Peter D. McLoone
Research Mentor: Michael Huang
July 30, 2009

What is OSIRIS?

OSIRIS is a fully explicit, multi-dimensional, fully parallelized, and fully relativistic code written in Fortran 90 and it models intense plasma beam interactions which are very complex and highly nonlinear. OSIRIS supports Cartesian geometry in 1D, 2D, and 3D and cylindrical geometry in 2D. External fields include electromagnetic waves that can be launched from a stationary antenna. Boundary conditions include periodic, conducting, and absorbing for fields and periodic, reflecting, and thermal bath for particles. It has a plasma ionization package, including both field and impact ionizations, and also a collision model. Additional features include load balancing, higher-order spline current deposition, particle tracking diagnostics, and arbitrary initial density profiles. One of OSIRIS's strongest attributes is a sophisticated array of diagnostic and visualization packages[1].



Visualization of OSIRIS output

What is NVIDIA CUDA?

NVIDIA® CUDA™ is a general purpose parallel computing architecture that leverages the parallel compute engine in NVIDIA graphics processing units (GPUs) to solve many complex computational problems in a fraction of the time required on a CPU. It includes the CUDA Instruction Set Architecture (ISA) and the parallel compute engine in the GPU. To program to the CUDATM architecture, developers can, today, use C, one of the most widely used high-level programming languages, which can then be run at great performance on a CUDATM enabled processor[2].

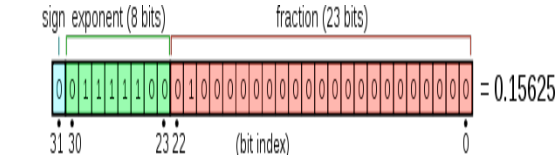
OSIRIS with CUDA

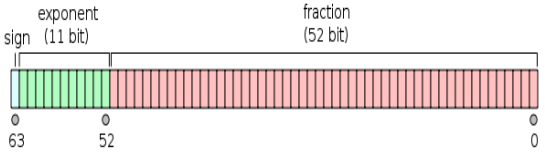
At U of R, portions of the OSIRIS v2 code have been ported to be executed on the GPU. The highly parallelized nature of the OSIRIS code lends itself well to the parallel units present on the graphics card. Presently, a finite difference solver for the electric and magnetic fields in both space and time and the mechanism for pushing the particles on the grid have been successfully implemented. Future additions include current smoothing across the grid of particles being simulated and sorting algorithms to enable further parallelization.

Single vs. Double Precision

Single and double precision refer to the amount of space in memory and definition of binary numbers in computer memory. They are currently defined by IEEE 754-2008, but in the past they have been defined simply by proprietary equipment. Typically, single and double precision numbers take up four and eight bytes, respectively. Schemes for the representations of the numbers are shown in the graphics. Single precision covers about seven decimal digits and double precision covers sixteen. In general cases, speed differences between the two are minimal.

However, in the case of using OSIRIS w/ CUDA, a possible difference arises. Double precision computational units on GPUs are relatively new and because they require more surface area on a chip there are fewer of them. This means that there is a potential for increased optimization beyond the inherently parallel nature of the GPU by taking code that was originally written in double precision for execution on a CPU and porting it to single precision.





Results and Future Work

At the present time, a single-precision version of OSIRIS w/ CUDA is not ready. That version of the code is in development and will likely be completed early in the fall. However, a single-precision version of OSIRIS w/o CUDA was produced in working towards a full version. At right is a table that compares some iterations of the code for a typical sample input. As expected, the single-precision code take approximately 2/3 of the time for the double precision – a minimal gain. The differences in rounding accrued by only having seven decimal digits instead of sixteen appear minimal.

Iteration	SP Time (s)	DP Time (s)	Error(%)
1	3.3562107	5.1043187	.34
10	3.4122124	5.1363210	.55
25	3.3042069	5.0043130	.71
50	3.2602081	4.9483089	1.7
75	3.2122193	4.8883059	.07
100	3.2241822	4.9003059	.64

It is expected that successful implementation of a single-precision version of the OSIRIS w/ CUDA code would produce a gain of severalfold.

References:
1. Fonseca R~A, Silva L~O, Hemker R~G, Tsung F~S, Decyk V~K, Lu W, Ren C, Mori W~B, Deng S, Lee S, Katsouleas T and Adam J~C 2002 Osiris: a three-dimensional, fully relativistic particle in cell code for modeling plasma based accelerators {\em Lect. Notes Comput. Sci.V} vol 2331 (Springer-Verlag) pp III--342:351

2 . http://www.nvidia.com/object/cuda_what_is.html
NVIDIA Corporation 27 July 2009

This research was funded by a generous gift from the Xerox Corporation