Using CUDA to Accelerate Analysis of Kerr Tails

Literature Review

John Dickerson

April 6, 2009

Overview

In an effort to facilitate an easy understanding of the project, this literature review is split into three sections. The first section is the most general, covering the physics and astronomy research that forms a basis for the problem at hand. Following this is a high-level overview of the parallel algorithms and mathematical concepts I use to solve the problem. Finally, the third section covers prior work done at the coding and implementation level with C for CUDA.

Prior Work in Physics and Astronomy on Late Time Behavior of Black Holes

The canonical work on late-time behavior of gravitational fields of black holes was published by Karl Schwarzchild in 1916 [18]. As a star collapses through its gravitational radius to form a black hole, it leaves behind a gravitational field. Schwarzchild's work studies the dynamics of perturbations in this gravitational field using scalar fields. His work focuses on black holes without any angular momentum or charge (known as Schwarzchild black holes), and thus he only studies *static* gravitational fields. Schwarzchild determines that any perturbations of a static black hole's gravitational field are themselves spherical. Phrased differently, the geometry of these gravitational fields is guaranteed to be spherically symmetric. Richard Price's additional work in the area recognizes this geometric symmetry and applies spherical harmonics to compactly describe the decay of these field perturbations [16].

A newer area of research involves a similar problem, only applied to the gravitational fields of Kerr black holes [12]. Kerr black holes, unlike their Schwarzchild counterparts, have nontrivial angular momentum, meaning they are rotating. It is believed that the center of the Milky Way contains a quickly spinning black hole of this type, GRS 1915+105, with a spin rate of around 1000 rotations per second [7]. Nontrivial angular momentum destroys gravitational fields' spherical symmetry and thus complicates the spherical harmonics-based analytical description with additional multipoles [11] [1] [3].

The analytical descriptions provided by two of these papers, Hod [11] and Burko [3], differ in their predictions. Hod enumerates a list of equations that describe late-time field decay, where an appropriate equation is chosen based on the initial pure multipole structure of the gravitational field (at time $t = 0$). Contrary to this, Burko's work claims fields decay in a manner similar to that of a Schwarzchild black hole's late time decay.

Work done by Manuel Tiglio at the University of Maryland explores if and when either case is correct [19]. Code written to simulate the decay of a Kerr black hole's gravitational field shows that, for low initial multipole values, Hod's predictions match Burko's. However, for higher initial multipole values, their predictions diverge. Furthermore, Tiglio's work supports Burko's model in the case of Kerr-Schild coordinates and Hod's model in the case of Boyer-Lindquist coordinates being used.

The calculations required to simulate the evolution of these perturbations are computationally intensive, as seen in Tiglio's and others' work [19] [3]. My goal is to leverage the power of consumer-level graphics cards to significantly decrease the execution time of these simulations. The next two sections describe, in detail, the numerical methods I aim to port to the graphics card as well as descriptions of the mathematical functionality and libraries already available on current-day graphics cards.

Numerical Methods, Specifically Differentiation and Integration

The code used to predict the evolution of perturbations on the gravitational field of a Kerr black hole can be simply described as the repetition of a function taking perturbation data at a time t and determining how it will change by time $t + 1$. This function, described at a high level by Tiglio [19], relies heavily on spectral methods for differentiation. These techniques are well-documented [2] and typically center around the use of the Fast Fourier Transform (FFT), an algorithm for quickly computing the Discrete Fourier Transform (DFT) and its inverse [4].

FFT lowers the computational complexity of computing a DFT on n points from $O(n^2)$ to $O(n \log n)$. Although not reducing the complexity further, computing an FFT in parallel provides significant real-world decreases in computation time. Further speedups can be achieved by computing an approximation to the DFT, in parallel, with a certain error bound [8].

Since graphics hardware relies on discrete, single-precision mathematical operations, some level of error will be inherent in any FFT implementation, whether exact or approximate. Finally, since double-precision floating point arithmetic is common with CPU-based FFT implementations, it will be important to compare the results of my single-precision GPU computations to the double-precision results of Tiglio's code.

Similar Problems Studied on GPUs, and Algorithm Implementations in C for CUDA

The manycore architecture of today's GPUs lends itself to speedups in parallelizable scientific computing problems. Furthermore, GPUs' stream processing capabilities are nicely applied to similar problems [9].

Nvidia Corporation provides both a low- and high-level API that allows for programming in a C-like language, dubbed C for CUDA, directly onto their newer (G80+) graphics hardware [10]. The numerical methods I am porting to the hardware will be written in this language. Since the original code on which I am basing this project is written in Fortran 90, I will use FLAGON, a Fortran 9x wrapper built by Ramani Duraiswami at the University of Maryland, to tie the two codebases together [15].

Nvidia provides an implementation of the Basic Linear Algebra Subprograms, called CUBLAS [5], as well as a parallel FFT solver called CUFFT [6]. I will begin by using the CUFFT library coupled with FLAGON. Comparisons of the results obtained from this library and those from "unofficial" sources [17] will follow.

Much work has been done regarding black hole simulation on parallel machines. For example, Makino simulated a Kerr black hole in a galactic center on the GRAPE-6 system, a supercomputer built specifically for particle simulation [14]. My research will be the first of its kind for a consumer-level GPU-based simulation of perturbations on the gravitational field of a black hole. It will complement more general GPU simulation research, like the n-body simulation on CUDA provided by Nyland [13]. As standard supercomputers can be prohibitively expensive, consumer-level hardware-based research like this brings scientific computing onto the affordable, ubiquitous personal computer.

References

[1] Leor Barack and Amos Ori. Late-time decay of scalar perturbations outside rotating black holes. *Phys. Rev. Lett.*, 82(22):4388–4391, May 1999.

[2] Guo Ben-Yu. *Spectral methods and their applications.* World Scientific, 1998.

[3] Lior M. Burko and Gaurav Khanna. Radiative falloff in the background of rotating black holes. *Phys. Rev. D*, 67(8):081502, Apr 2003.

[4] James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301, 1965.

[5] Nvidia Corporation. Cuda: Cublas library. September 2008.

[6] Nvidia Corporation. Cuda: Cufft library. October 2008.

[7] Andreas Eckart, Reinhard Genzel, and Rainer Schödel. The massive accreting black hole at the center of the milky way *. *Progress of Theoretical Physics Supplement*, 155:159–165, 2004.

[8] Alan Edelman and Peter Mccorquodale. The future fast fourier transform. *SIAM J. Sci. Computing*, 20:1094–1114, 1999.

[9] Jayanth Gummaraju and Mendel Rosenblum. Stream processing in general-purpose processors.

[10] Tom Halfhill. Parallel Processing with CUDA. *Microprocessor Journal*, 2008.

[11] Shahar Hod. Mode coupling in rotating gravitational collapse: Gravitational and electromagnetic perturbations. *Phys. Rev. D*, 61(6):064018, Feb 2000.

[12] Roy P. Kerr. Gravitational field of a spinning mass as an example of algebraically special metrics. *Phys. Rev. Lett.*, 11(5):237–238, Sep 1963.

[13] Jan Prins Lars Nyland, Mark Harris. *Fast N-Body Simulation with CUDA*. Addison-Wesley Professional, 2007.

[14] Junichiro Makino, Toshiyuki Fukushige, and Masaki Koga. A 1.349 tflops simulation of black holes in a galactic center on grape-6. In *Supercomputing '00: Proceedings of the 2000 ACM/IEEE conference on Supercomputing (CDROM)*, page 43, Washington, DC, USA, 2000. IEEE Computer Society.

[15] W. Dorland N. Gumerov, R. Duraiswami. Middleware for programming nvidia gpus from fortran 9x. *Supercomputing 2007.*

[16] Richard H. Price. Nonspherical perturbations of relativistic gravitational collapse. ii. integer-spin, zero-rest-mass fields. *Phys. Rev. D*, 5(10):2439–2454, May 1972.

[17] Sergio Romero, Maria A. Trenas, Eladio Gutierrez, and Emilio L. Zapata. Locality-improved fft implementation on a graphics processor. In *ISCGAV'07: Proceedings of the 7th WSEAS International Conference on Signal Processing, Computational Geometry & Artificial Vision*, pages 58–63, Stevens Point, Wisconsin, USA, 2007. World Scientific and Engineering Academy and Society (WSEAS).

[18] K. Schwarzschild. On the gravitational field of a sphere of incompressible fluid according to einstein's theory. February 1916.

[19] Manuel Tiglio, Lawrence Kidder, and Saul Teukolsky. High accuracy simulations of kerr tails: coordinate dependence and higher multipoles. 2007.