Integrating Post-Newtonian Equations on Graphics Processing Units

F. Herrmann[1,2,3], J. Silberholz[1,4,5], G. Guerberoff[6], M. Tiglio[1,2,3]

University of Maryland: [1] Center for Scientific Computation and Mathematical Modeling, [2] Department of Physics, [3] Center for Fundamental Physics, [4] Department of Mathematics, [5] Department of Computer Science
[6] Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay

Summary

The merger of 2 black holes is expected to be a common event in the universe. We perform a statistical (Monte-Carlo) analysis of an approximate solution to the binary black hole inspiral problem to find preferred system states before merger.
Using GPUs we achieve a speed-up of a factor 50 over a CPU solution making a large scale study feasible on the NCSA Lincoln cluster with 96 Tesla S1070 compute units.
Details: http://arxiv.org/abs/0908.3889

Black Hole (BH) Mergers

Orbiting black holes lose energy and angular momentum to gravitational radiation. This shrinks the orbit and eventually merger occurs. During merger up to 10% of the rest mass are radiated making this the brightest event in the universe.

Post-Newtonian Approximation

The Post-Newtonian approximation is a series expansion of Einstein's General Relativity valid for slowly-moving, far-separated objects. For circular inspirals one obtains a system of coupled ordinary differential equations (ODEs) for the variables: orbital frequency ω, the individual spin vectors $\mathbf{S}_i$ for the 2 BHs, and the unit orbital angular momentum vector $\hat{\mathbf{L}}_n$.

Post-Newtonian Equations

$$\begin{aligned}
\dot{\omega} &= \omega^2 \frac{96}{5}\eta (M\omega)^{5/3} \Bigg\{ 1 - \frac{743+924\eta}{336}(M\omega)^{2/3} \\
&- \left(\frac{1}{12}\sum_{i=1,2}\left(\chi_i \hat{\mathbf{L}}_n \cdot \hat{\mathbf{S}}_i \left(\frac{113 m_i^2}{M^2} + 75\eta\right)\right) - 4\pi \right) M\omega \\
&+ \left(\frac{34103}{18144} + \frac{13661}{2016}\eta + \frac{59}{18}\eta^2\right)(M\omega)^{4/3} \\
&- \frac{1}{48}\eta\chi_1\chi_2 \left(247(\hat{\mathbf{S}}_1\cdot\hat{\mathbf{S}}_2) - 721(\hat{\mathbf{L}}_n\cdot\hat{\mathbf{S}}_1)(\hat{\mathbf{L}}_n\cdot\hat{\mathbf{S}}_2)\right)(M\omega)^{4/3} \\
&- \frac{1}{672}(4159+15876\eta)\pi(M\omega)^{5/3} \\
&+ \Bigg(\left(\frac{16447322263}{139708800} - \frac{1712}{105}\gamma_E + \frac{16}{3}\pi^2\right) + \left(-\frac{273811877}{1088640} + \frac{451}{48}\pi^2 - \frac{88}{3}\hat{\theta}\eta\right)\eta \\
&+ \frac{541}{896}\eta^2 - \frac{5605}{2592}\eta^3 - \frac{856}{105} log(16(M\omega)^{2/3}) \Bigg)(M\omega)^2 + \left(-\frac{4415}{4032} + \frac{358675}{6048}\eta + \frac{91495}{1512}\eta^2\right)\pi (M\omega)^{7/3} \Bigg\} \\
\dot{\mathbf{S}}_i &= \mathbf{\Omega}_i \times \mathbf{S}_i \\
\dot{\hat{\mathbf{L}}}_n &= -\frac{(M\omega)^{1/3}}{\eta M^2}\frac{d\mathbf{S}}{dt}
\end{aligned}$$

and $\mathbf{\Omega}_1$ given by: $\mathbf{\Omega}_1 = \frac{(M\omega)^2}{2M}\left(\eta (M\omega)^{-1/3}(4 + 3\frac{m_2}{m_1})\hat{\mathbf{L}}_n + 1/M^2(\mathbf{S}_2 - 3(\mathbf{S}_2\cdot\hat{\mathbf{L}}_n)\hat{\mathbf{L}}_n)\right)$

Parallel ODE Integration

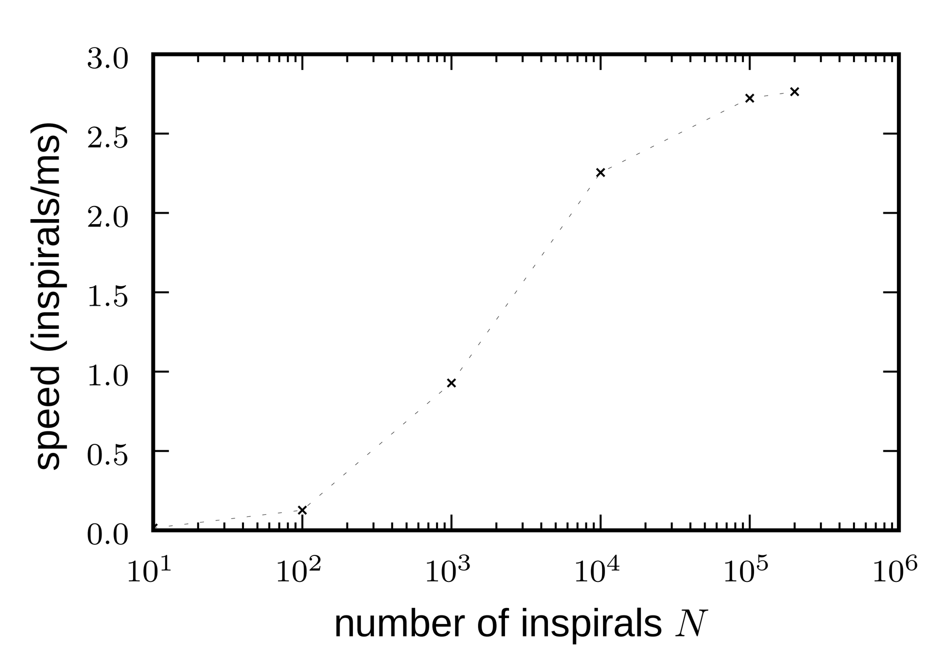
To integrate the Post-Newtonian equations we use a standard adaptive time-stepping ODE algorithm: Dormand-Price. We choose the initial conditions for the ODE integrations randomly. We then spawn CUDA kernels to perform as many of the integrations in parallel as possible.
Each integration uses its own adaptive time step. For this problem the vast majority of inspirals can be performed with a time-step of similar size until very close to the end of integration. This makes potential load-balancing issues over cores much simpler.

```
// compute right hand side (rhs) of ODE
__global__ void ode_rhs(state)

// integrate the ODEs
void integrate(initial_state_and_params) {
  allocate_gpu_storage(...);
  while (all_omega>all_omega_final) {
    ode_rhs<<<nBlocks_rhs,nThreads>>>(...);
    checkCUDAError("first ode_rhs call");
    interm_1<<<nBlocks_interm,nThreads>>>(...);
    ... // more intermediate values & rhs calls
    adjust_timesteps(...);
  }
  transfer_from_gpu_to_host(...);
}
```

Pseudocode describing the parallel ODE integration.

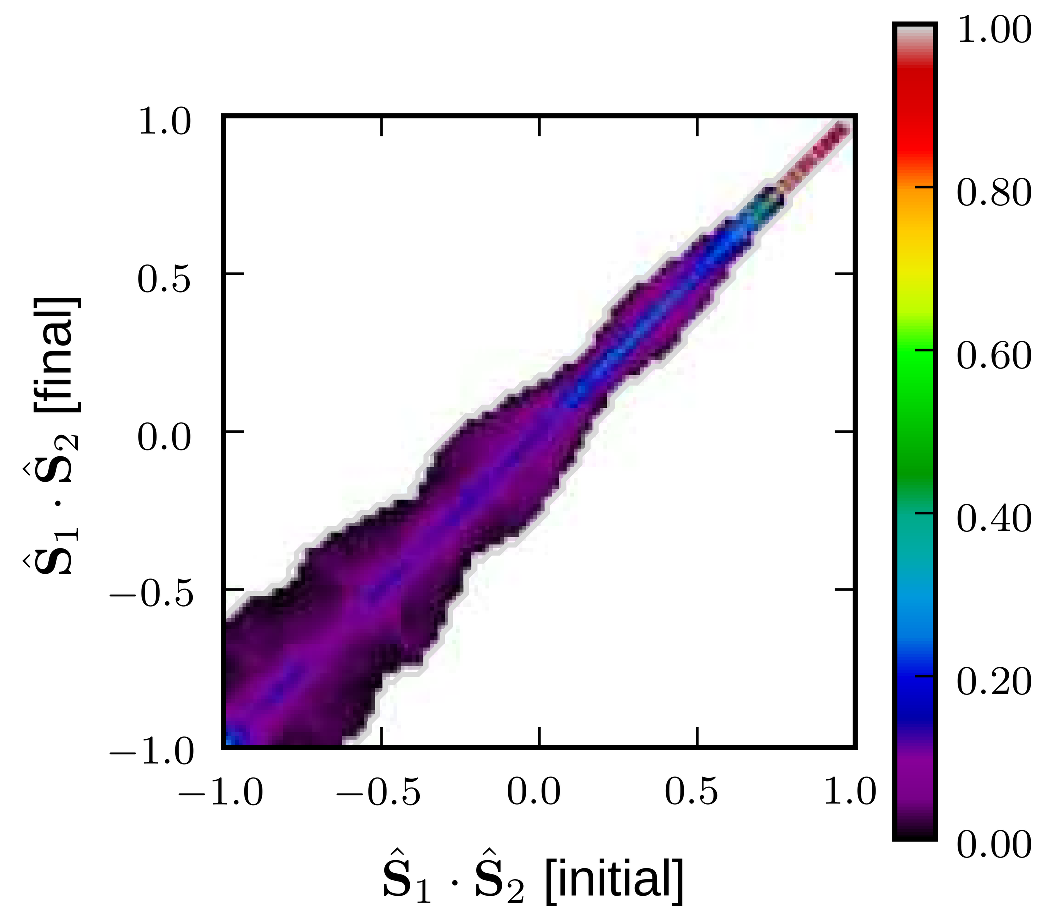
Performance



The plot shows the number of test runs per millisecond that can be performed on the card as the number of runs N is increased. The performance levels off at around 2.6 inspirals/ms for $N \geq 100,000$. The Tesla card we used in this study has 240 cores and the CUDA runtime can schedule the extra threads efficiently which results in performance improvement of a factor 50 over a single core Xeon E5410 CPU in double precision.

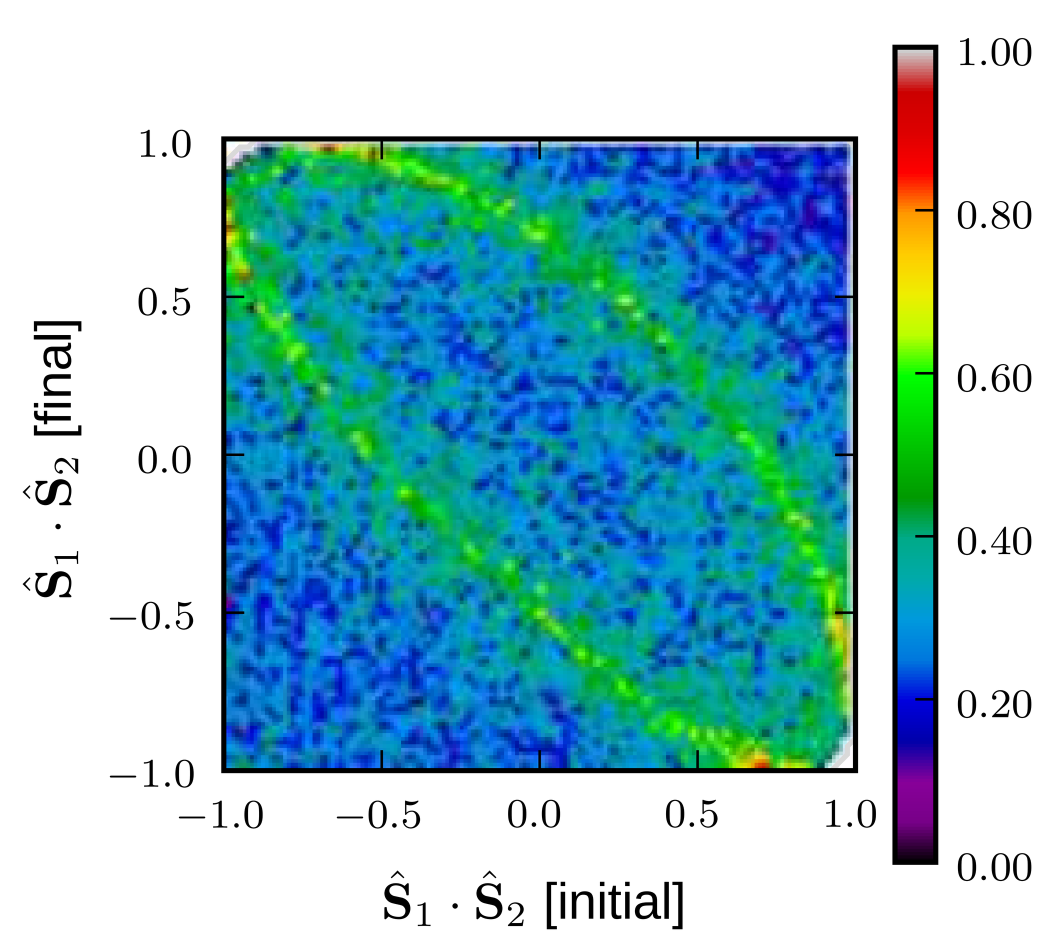
Equal-mass, maximally spinning BHs

If the 2 BHs have equal mass $m_1 = 0.5$ and are maximally spinning $\chi_1 = \chi_2 = 1$ the system behaves predictively and a high level of correlation in the scalar product of initial and final spin vectors remains.



Unequal-mass, low-spin BHs

If one moves away from the equal-mass case and chooses fairly low spin magnitude then a richer structure appears. The parameters chosen for this figure are $m_1 = 0.4$, $\chi = 0.05$. This case has sensitivity on the initial and final orbital frequencies ω_i, ω_f.



Conclusions

- We have implemented a parallel ODE integrator in the NVIDIA CUDA environment and integrated the post-Newtonian equations of motion describing the inspiral of two black holes.
- GPUs provide an excellent environment for the parallel integration of ODEs giving substantial speed-ups over CPUs.
- We next plan to extend the initial studies performed here to large-scale studies of the 7-dimensional inspiral phase-space with production runs currently undertaken on the NCSA Lincoln GPU cluster.

Further Information

Two talks will be presented at this conference:

1. John Silberholz. Session List 1441.
   Friday, California Room, 1:30pm-2:00pm
2. Frank Herrmann. Session List 1402.
   Friday, California Room, 2:00pm-2:30pm

Please see our paper [1]. A preprint is available at:

http://arxiv.org/abs/0908.3889

For follow-up questions feel free to contact us at

Frank Herrmann	fherrman@umd.edu
John Silberholz	josilber@umd.edu
Gustavo Guerberoff	gguerber@gmail.com
Manuel Tiglio	tiglio@umd.edu

Acknowledgments

Work supported by NSF grant PHY0801213 and an NVIDIA professor partnership. Simulations were done using 2 NVIDIA Tesla S1070 Computing Systems at UMD as well as the NCSA Lincoln cluster.

References

[1] F. Herrmann, J. Silberholz, M. Bellone, G. Guerberoff, M. Tiglio. *Integrating Post-Newtonian Equations on Graphics Processing Units.* http://arxiv.org/abs/0908.3889