Homework 5: Monte Carlo on GPUs

Due June 3rd.

1. Write a (non-GPU) program to calculate π by integrating the area of a circle of radius 1 with the Monte-Carlo method. You may use a system-provided random number generator. The Python `random` module says it uses the Mersenne Twister which I mentioned in class as a quality generator. Plot how the error in the estimated value of π changes with the number of points used to evaluate the integral; does this follow the expected error for a Monte Carlo method?

2. Implement the same Monte-Carlo method using CUDA on a CUDA capable machine. If you are using the Astronomy Linux systems, "magneto" is a machine with dual Quadro FX 5600s. This is a somewhat older GPU that only supports 32 bit floating point (float32). I have installed the NVIDIA compiler in `/astro/users/trq/cuda-5.5` with the compiler in `bin` and libraries in `lib64` subdirectories. Note that you may need to adjust your `LD_LIBRARY_PATH` to point at the `lib64` directory. I recommend using the random number generator in the CUDA library (e.g., `curand_uniform()`). For those familiar with Python, this may be easier using PyCUDA. I have installed this on the Astronomy machines in `/astro/users/trq/python/lib` using the `/astro/apps6/bin` version of Python. Verify that you get an accurate estimate of π, and compare the time taken with the CUDA implementation with the time taken by the non-CUDA implementation.