cuRAND

cuRAND uses GPU to generate pseudorandom numbers

Host and Device API

Host API like cuBLAS but for random numbers

Device API can generate numbers while in kernel, more complicated

Monte Carlo Example

Cuda with random numbers is very useful for Monte Carlo

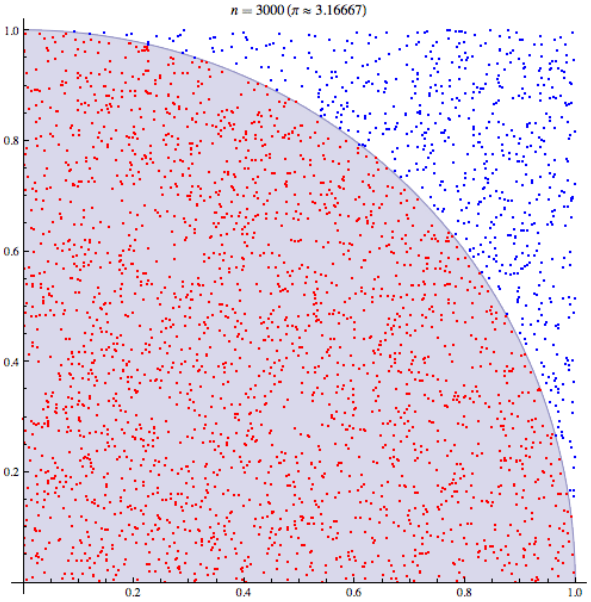
Repeated random sampling of the same system to get results

Each repetition is independent -> parallelizable!

Monte Carlo Example

Simple example: numerically approximate pi

randomly scatter points, in a region, and calculate the fraction of points inside the circle



Monte Carlo Example

Parametric definition of a circle makes it easy to tell which points are in the circle

x^2 + y^2 < 1 : inside unit circle

Monte Carlo Example

Generate points uniformly distributed in {(0, 1], (0, 1]}

```
float* dev_points;
curandGenerator_t gen;
curandCreateGenerator(&gen,CURAND_RNG_PSEUDO_DEFAULT);
curandGenerateUniform(gen, dev_points, numPoints * 2);
```

Monte Carlo Example

Check bounds and reduce

```
__global__ void countPointsInCircle(float* points, int n)
{
    extern int sharedMem[];
    int idx = threadIdx.x + blockIdx.x * blockDim.x;
    float x = points[idx];
    float y = points[idx + n];
    sharedMem[threadIdx.x] = (x * x + y * y < 1);
    //syncthreads and perform summing reduction
}
```

Monte Carlo Example

Just use our count to calculate the value we want

sum/numPoints is pi/4

Gillespie algorithm

Stochastic simulation of dynamic systems

Individual transitions are memoryless, if nothing happens for a long time, the system does not “remember” this

Gillespie algorithm

Simulation step:

Calculate the probability of each transition being the next event, calculate the distribution of the time the next event will occur at

Gillespie algorithm

Minimum of exponential random variables

if X1, X2, X3… are exponential random variables

min(X1, X2, X3…) is an exponential random variable with $\lambda=\lambda1 + \lambda2 + \lambda3...$

Gillespie algorithm

So we can calculate our transition distributions

P(event i happens next) = λi/sum(all λ)

Distribution of times

T : exponential with rate constant = sum(all λ)

Gillespie algorithm

We can sample these random variables using uniform distributions taken from cuRAND

Simple Chemical Simulation

A biological molecule is produced by a large reservoir of reactants, and decays at a rate proportional to concentration

reactants -> X -> 0

Simple Chemical Simulation

Nonstochastic model

$d[X]/dt = k - \gamma * [X]$

Only accounts for average behavior
What is the variance of [X] at equilibrium?

Simple Chemical Simulation

Reaction propensities

X++	k
X--	γ * X

Simple Chemical Simulation