Particle Simulations in CUDA

Stephen Macke and Rachel Turner
University of Tulsa, Youngstown State University
stephen-macke@utulsa.edu, raturner@ysu.edu

Introduction

Particles from emission and other processes undergo physical events, such as coagulation. The atmospheric consequences of these events are of special interest, as they may contribute to climate change, or negatively impact human pulmonary health.

Because of the inherent difficulty in collecting real-world data for these processes, they are often simulated using computer models. These models are typically executed using many thousands of particles.

Typical implementations suffer from one of the following:

1. Computation time, which scales with the number of particles, becomes prohibitively slow on personal computers.
2. It is expensive to alleviate this performance issue with supercomputers.

GPU computing offers an affordable alternative to supercomputers for massively parallel applications. Our main contribution is the use of CUDA to speed up routines for the following in existing **P**article-resolved **M**onte **C**arlo (PartMC) code for atmospheric aerosol simulation:

1. Condensation of water vapor on aerosol particles
2. Aerosol particle coagulation

Condensation

Particle condensation is the accumulation of water vapor on aerosol particles. This ultimately affects the particle wet diameter. As humidity increases, the amount of water condensing on particles increases and the diameter of each particle expands. Condensation is a contributing factor of cloud formation. The PartMC condensation code simulates the change in diameter of particles due to condensation. Since large numbers of particles are needed to get useful results it is important that this code run faster and more efficiently. Parallelizing this condensation code was very straight forward. The set of wet diameters is governed by a system of ODE's (given in [1]), and the method the ODE solver uses (a backwards difference method) requires expensive, independent Newton iterations for each particle. Since each particle does not communicate with any other particle, the CUDA parallelization of these routines was trivial.

Coagulation

Particle coagulation is the process whereby aerosol particles collide and stick together, effectively forming a new, larger particle. Any two particles have a certain probability per unit time of coagulating. This probability is determined by the value of a kernel type, which depends on properties of the two colliding particles.

Below is a plot of the Brownian kernel, which gives a fairly good approximation for what actually occurs in nature.

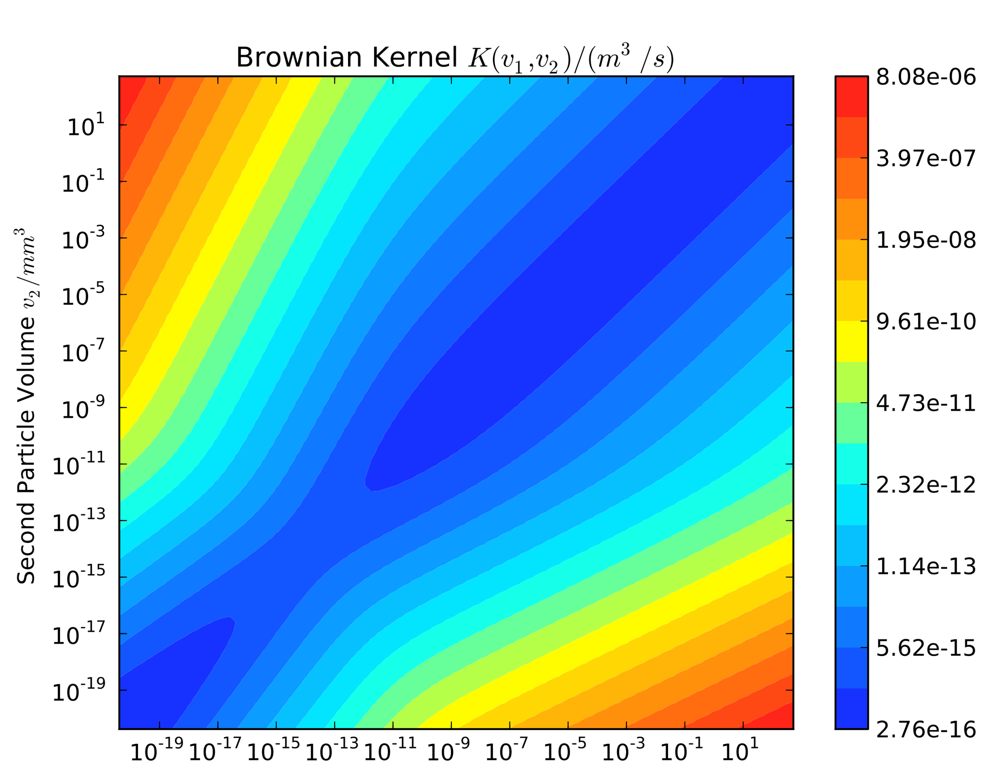


Figure 1: The kernel takes on larger values for particles with very different volumes. For simplicity, all the particles in the plot have the same density.

The kernel is a function from $\mathbb{R}^4$ to $\mathbb{R}$, as it depends on the volumes and densities of two separate particles.

Coagulation: Serial Tau Leaping

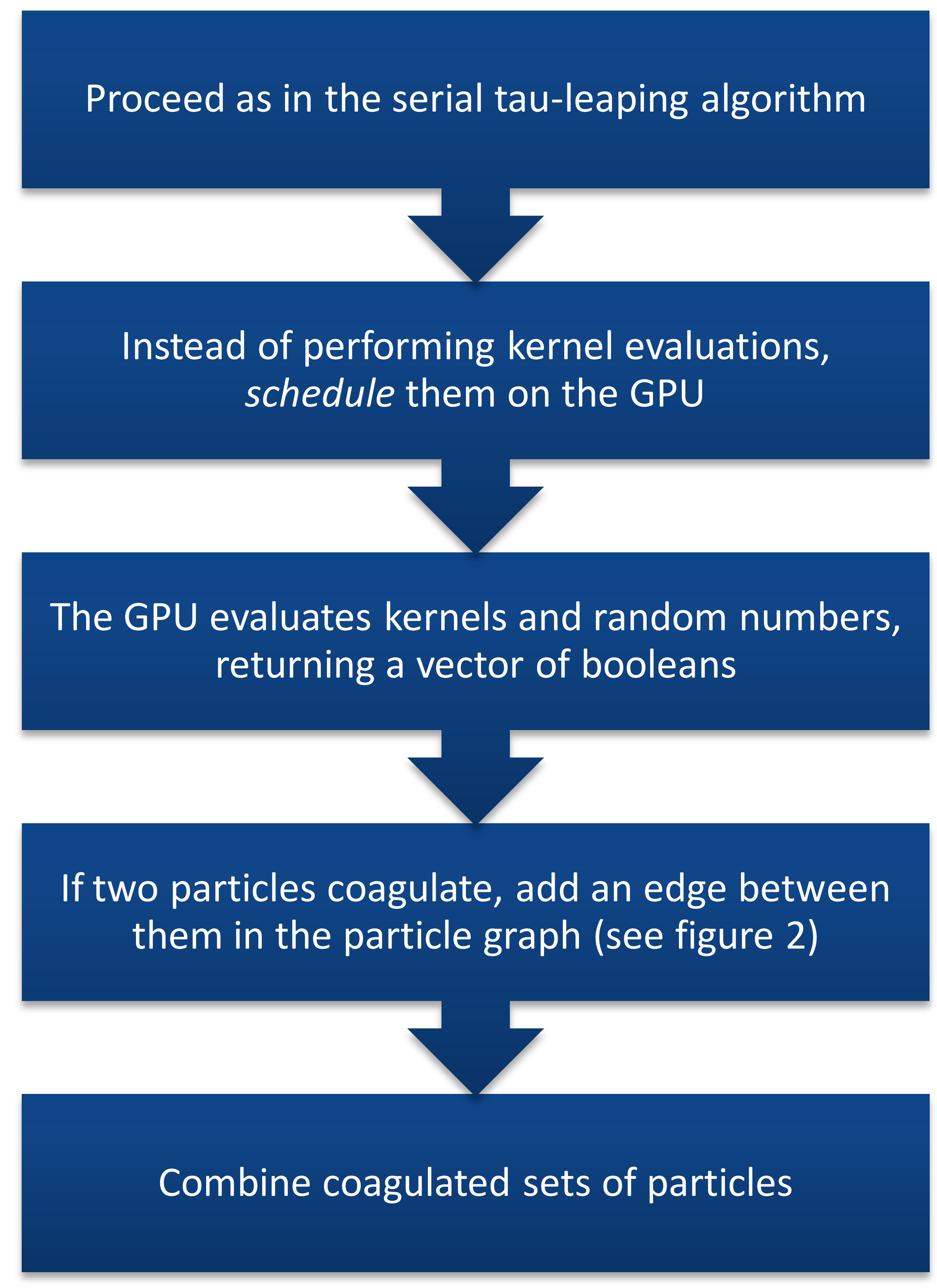
Suppose the simulation involves N_p particles. To avoid testing all $N_p \cdot (N_p - 1)$ particle pairs for coagulation, PartMC stochastically samples particle pairs, using a binned accept-reject approach for efficiency purposes.

Below is pseudocode for this binned tau-leaping method for coagulation, similar to what is given in [2]:

divide diameter axis into bins, on a log scale
V is the computational volume
$N_{\mathrm{p}}(b)$ is number of particles in bin b
$\vec{\mu}(b, i)$ is mass vector of the i-th particle in bin b
$K_{\max}(b_1, b_2)$ is a precomputed upper bound on the kernel for any particles from bins b_1 and b_2
Δt is the timestep

for all bin pairs (b_1, b_2) with $b_1 \le b_2$ **do**
  $N_{\mathrm{event}} \leftarrow N_{\mathrm{p}}(b_1)(N_{\mathrm{p}}(b_2) - \delta_{b_1,b_2})/(1 + \delta_{b_1,b_2})$
  $N_{\mathrm{test}} \leftarrow \mathrm{Pois}(K_{\max}(b_1, b_2)\Delta t N_{\mathrm{event}}/V)$
  for N_{test} repetitions **do**
    randomly choose particles i_1 and i_2 uniformly in bins b_1 and b_2
    $K_{12} \leftarrow K(\vec{\mu}(b_1, i_1), \vec{\mu}(b_2, i_2))$
    randomly choose r uniformly in $[0, 1]$
    if $r < K_{12}/K_{\max}(b_1, b_2)$ **then**
      coagulate the two particles, updating the arrays $N(b)$ and $\mu(b, i)$
    end if
  end for
end for

Parallel Tau Leaping



Maintaining Sets of Coagulating Particles

In a single time step, one particle may coagulate with multiple other particles (usually a large particle with several smaller ones – this is more likely, as evident in figure 1). Keeping track of these sets is essentially the same problem as adding edges to an undirected graph and maintaining connected components, for which a union-find structure is most appropriate (though in practice, we use a slightly more complicated technique).

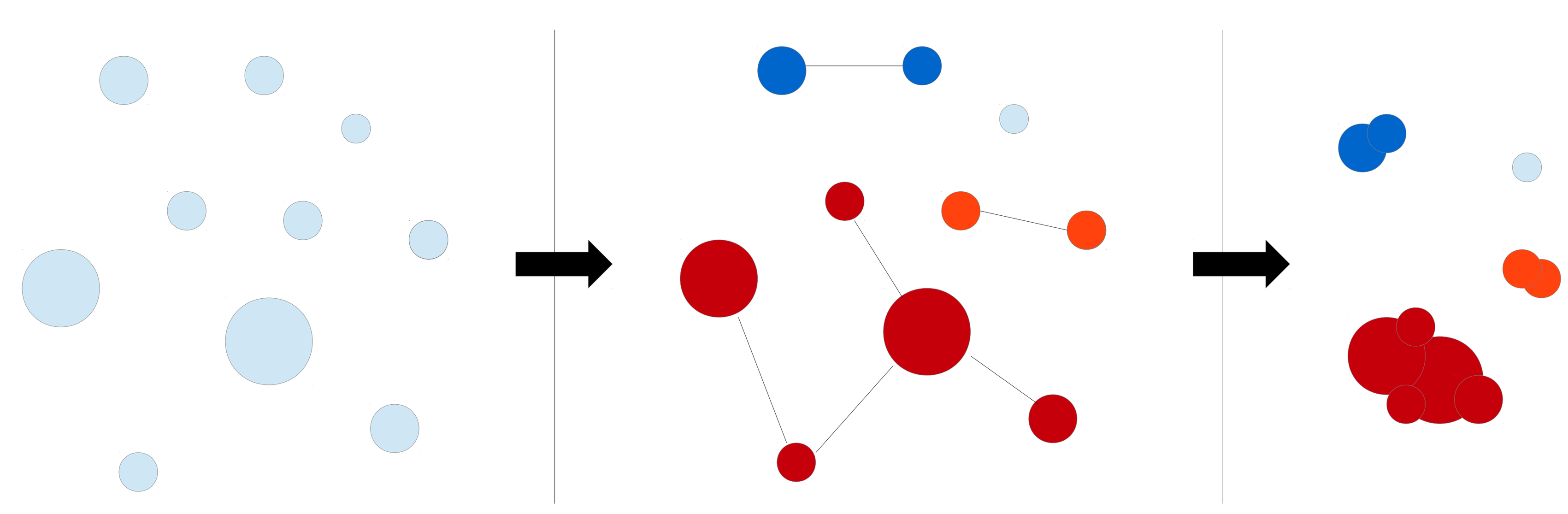
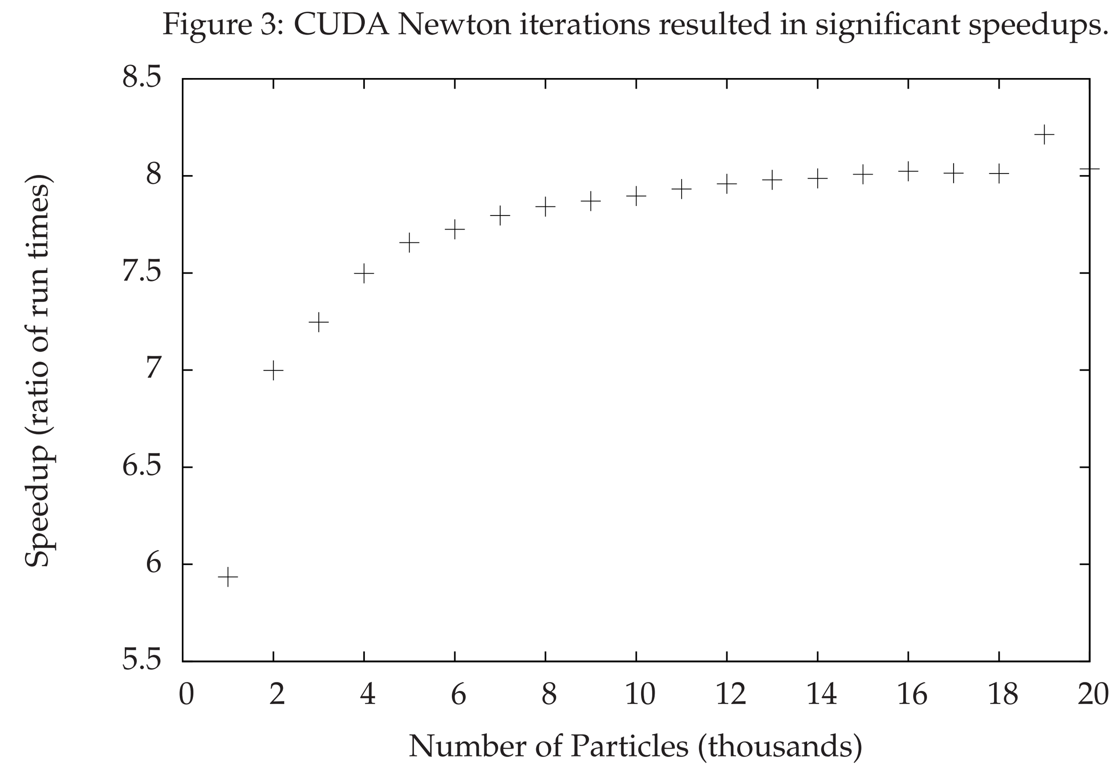


Figure 2: Determining coagulations in parallel. After pairwise coagulations are determined from kernel values computed on the graphics card, a union-find algorithm (performed on the CPU) maintains connected components in a forest of disjoint trees. These connected components are then combined.

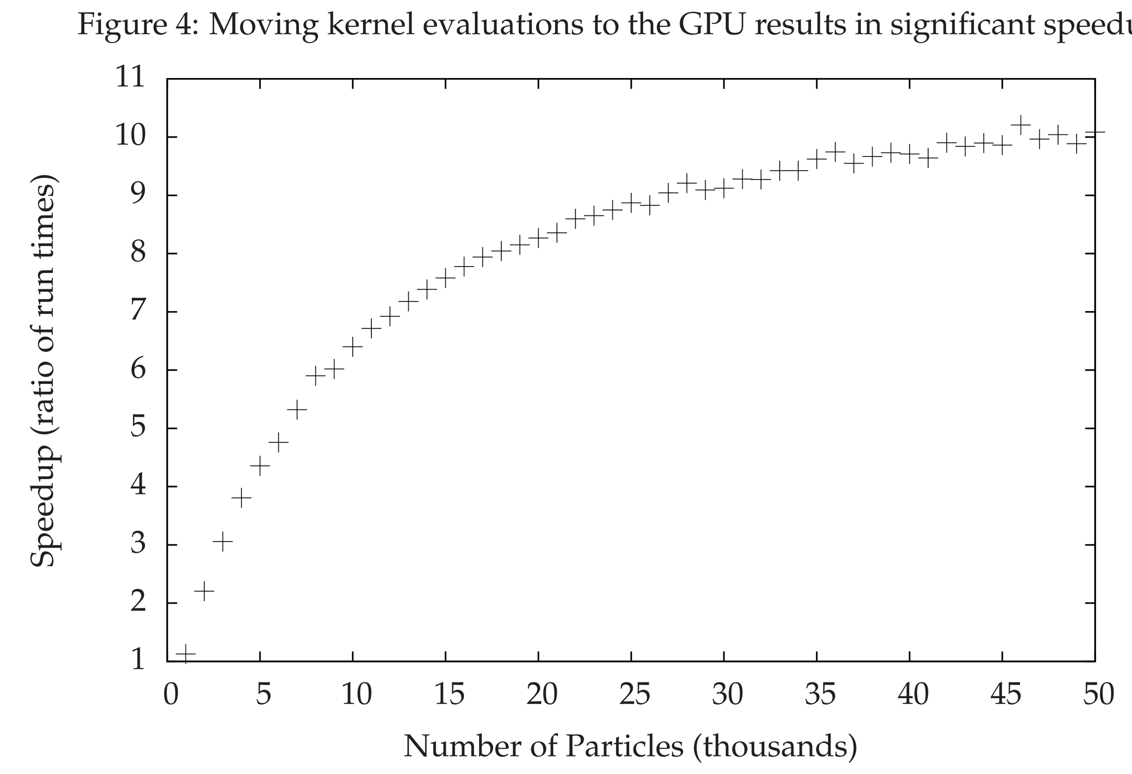
Condensation Results

Figure 3: CUDA Newton iterations resulted in significant speedups.



Coagulation Results

Figure 4: Moving kernel evaluations to the GPU results in significant speedups.



References

[1] J. Ching, N. Riemer, M. West Impacts of black carbon mixing state on cloud droplet activation – Insights from a particle-resolved model *J. Geophys. Res., to appear* D09202, doi:10.1029/.

[2] N. Riemer, M. West, R. A. Zaveri, R. C. Easter, Simulating the evolution of soot mixing state with a particle-resolved aerosol model, *J. Geophys. Res., 114* D09202, doi:10.1029/2008JD011073.

Acknowledgements

We gratefully acknowledge support from NSF grants 1004311 and 0934491. Additionally we acknowledge valuable mentorship and assistance from Matthew West, Jeffrey Curtis, Matthew Michelotti, Craig Zilles, and Jill Peckham.

Passionate on Parallel
2012 REU Program

parallel COMPUTING INSTITUTE

NSF

This REU is co-funded by the ASSURE program of the Department of Defense in partnership with the National Science Foundation REU Site program under Award No. 1004311.