CS240a Project Proposal

Josh Novins and Sunimal Edussuriya
Professor Gilbert

For his research, Josh is studying the propagation and attenuation of directed lasers through the atmosphere in near-ocean situations. Ideally for a given set of boundary conditions the laser's attenuation over distance will be determined by considering the effects of temperature (or alternatively density) gradients as well as the effect of particle droplets in the air. This research assumes that the laser does not interact with the fluid/particle field; this part of the problem is uncoupled. However, generating the flow fields will be the bottleneck for research: the flow field and the particle motion is strongly coupled, and running such a code is not fast. For the purposes of the research, it would be very helpful if a parallelized code could be made that could generate a flow field in a reasonable amount of time.

There are several immediate avenues for parallelization. To start out with in principle it should be possible to easily parallelize the laser propagation. As the laser travels through the atmosphere, it's attenuation over a given stretch is independent of the future or past. It would be as simple as running a distributed memory code where each processor is assigned a given stretch of the laser's path. However, this portion of the research is negligible in terms of time.

The enormous bulk of the computational cost comes from simulating the particle-laden flow field. There are several avenues for parallelization - depending on how the code is written, the simulation of the flow field itself can be parallelized by breaking it into smaller chunks as we did with the matrix-vector multiplication earlier in the course. However, there are some difficulties with the flow field that need to be considered - the advection term for our simulation is not necessarily pointing in the same direction at all points, and the velocity is certainly different everywhere. This means that the amount of information needed in each subchunk of the simulation needs to be handled within each time step; in practice, each time step is of variable size, where a maximum time step is set so that no needed information isn't passed from one processor to the next. Even beyond this parallelization of the fluid equations, the particle motion in each time step can be independent. This too is tricky though - if the particles are not going to touch, then parallelization is simple, but if the particles touch then the particles themselves are going to influence each other's motion.

Ideally, we would write a parallelized code that can handle this simulation. However, this is likely a problem too large for the scope of the class. There is a code in my group looking at a similar problem that has been somewhat parallelized; our hope is that we can adapt this code and optimize/parallelize it for Josh's particular problem. If this proves to also be a very large problem, the minimum goal would be to make a 2d parallelized fluid solver without particles. Although not ideal, some 80% of laser attenuation is from temperature differences; having such a code would allow for large inroads to be made into the problem and could in the future be built upon to include particle motion.