CS 5220: Project Proposal

Lara Backer

October 7, 2015

1 Background

The development of a realistic combustion modeling capability is necessary to optimize and predict the behavior of alternative fuels that have the potential to reduce emissions and improve fuel efficiency. While tremendous strides have been made recently in modeling the turbulent spray combustion found in diesel and jet engines and gas turbines, a comprehensive understanding of the process is still lacking due to the immense computational cost of simulating the chemistry of complex fuels. In order to integrate detailed chemistry into simulations, additional scalar transport equations must be solved; the computational cost scales linearly with the number of species involved. Furthermore, the coupling between turbulence and combustion results in different time scales necessary to solve the flow equations [1]. An implicit scheme for integrating differential equations and computing the reaction source terms for the chemistry evolution ensures stability and allows a larger time step to be used, reducing computational cost.

From cursory simulations using NGA, a CFD code written in Fortran 90 [2] and a reduced chemical mechanism of approximately 150 species, the chemistry solver takes up 98% of the entire simulation runtime, even including velocity, pressure, and spray droplet solvers. Thus, more aggressive methods must be taken to reduce the time for the chemistry solver to make these simulations tractable.

2 Project Goal

The proposed project aims at reducing the computational cost of computing the chemical source terms, a process that takes up a significant portion of the chemistry solver. Currently, the simulation domain is partitioned into sections that each processor solves. In the case of spray ignition where a flame evolves non-uniformly in the domain, this can result in unbalanced loading of each processor for the chemistry step: one processor may contain the entire flame, while others may not contain any part of it, and thus do not have to do any chemistry computations whatsoever.

To achieve better load balancing, a dynamic master-slave algorithm is proposed to further divide each of the processor domains and package each of matrices of data and the subdomain chemistry computations off to any idle processors. This algorithm will be written using OpenMPI and Fortran 90, and will be entirely integrated into the NGA code. Simulations with and without the load balancing algorithm will be performed to contrast the effect of this decomposition on computational cost.

References

[1] Lu T, Law CK. Progress in Energy and Combustion Science. Progress in Energy and Combustion Science. 2009; 35(2):192-215.

[2] Desjardins O, Blanquart G, Balarac G, Pitsch H. High order conservative finite difference scheme for variable density low Mach number turbulent flows. Journal of Computational Physics. 2008; 227(15):7125-7159.