Project Proposal

▪ **Names of team members:**

Jonathan Lijewski, Nikodem Gazda

▪ **Project title:**

A Study and Analysis of High-Order Euler Equation Solver LAGHOS

▪ **Abstract (1 page or less) describing the selected application (from your preliminary reading of their publication or website:**

LAGHOS is a miniapp that is part of a larger software suite, CEED (Center for Efficient Exascale Discretizations), which is created and maintained by the US Department of Energy. All of the software miniapps included in CEED are used to create discrete models of high order, continuous models. LAGHOS in particular is used to describe high order Euler equations of gas dynamics in a Lagrangian frame into a discrete format. Given a system of high order differential equations, LAGHOS will construct matrices for the velocity, energy, and force of the system and solve them to create a mesh of the resulting discrete system.

LAGHOS contains two main modes for completing the discretization, partial assembly and full assembly. Partial assembly involves taking advantage of a sum factorization method and the local version of the matrices to precompute entries into the resultant matrix, saving in data storage and memory transfers, which is optimal when working with high order differential equations. The full assembly in comparison works with the global version of the matrices, and is contained in compressed sparse row format. Both methods utilize the MFEM C++ library to complete the calculations and construct the finite element meshes. Support is included for VisIt, a mesh visualization tool used to view the results of the LAGHOS software. In addition to the MFEM library, other external dependencies include hypre, a library used to parallelize the individual matrix calculations. LAGHOS maintains support for GPUs and additional software models such as CUDA and OpenMP through the MFEM library.

Approach (as much as you know now):

• **From where are you downloading the code (URLs of websites, etc.)?**

[Proxy Apps | Advanced Simulation and Computing (llnl.gov)]

• **What code are available? e.g., serial, MPI only, MPI+OpenMP, cuda, etc.**

Serial and MPI

• **What code are you planning to download port to HiPerGator and study/modify?**

Both the serial and MPI code

• **What are your proposed tasks (again as much as you know now)? E.g., perform analyses on the development of code; profile the downloaded code on HiPerGator and analyze the benchmarked data; compared to the published data, etc.**

- Read relevant works published on the solver and synthesize their findings
- Read through the downloaded code and write a detailed analysis of all its features, including how and why it was parallelized
- Port the code to Hipergator
- Compare the ported code to other studies done on LAGHOS

• **Any value-added tasks? E.g., modify code, design-space exploration (varying different system or application parameters).**

- Profile the code
- Modify the code and analyze how our modifications improve/worsen its performance
- Create test of a small-scale mesh using MPI + OpenMP code

▪ **Team members expertise and work distribution.**

- We both have experience with C++ from PCA as well as from other classes. We've both also learned MPI through PCA.
- It's likely we'll split the work load better as we work on the project more, but this is a guide:
  - We both find 2 papers each and synthesize their findings into our report
  - We both read through the code in its entirety, then split the code base in half to explain its features
  - One person compares the ported code results to other studies
  - One person profiles the code
  - We both modify sections of code and run it together, then split the explanation of what we did and its effects

▪ **Proposed schedule for the remainder of the semester.**

We'll set goals for every week and check in at the end of each week to see if we need to make changes to the schedule.

Nikodem: I'll be working primarily on Tuesdays and Sundays

Jon: I'll be working primarily on Mondays and Tuesdays

(3/25-4/3) - Research into LAGHOS and porting into HiperGator.

(4/4-4/13) - Profiling/Troubleshooting + work on progress slides

(4/14 - 4/28) - Value added tasks + final presentation

▪ **References and links: (at least 3)**

- Proxy Apps | Advanced Simulation and Computing (llnl.gov)
- Efficient exascale discretizations: High-order finite element methods - Tzanio Kolev, Paul Fischer, Misun Min, Jack Dongarra, Jed Brown, Veselin Dobrev, Tim Warburton, Stanimire Tomov, Mark S Shephard, Ahmad Abdelfattah, Valeria Barra, Natalie Beams, Jean-Sylvain Camier, Noel Chalmers, Yohann Dudouit, Ali Karakus, Ian Karlin, Stefan Kerkemeier, Yu-Hsiang Lan, David Medina, Elia Merzari, Aleksandr Obabko, Will Pazner, Thilina Rathnayake, Cameron W Smith, Lukas Spies, Kasia Swirydowicz, Jeremy Thompson, Ananias Tomboulides, Vladimir Tomov, 2021 (sagepub.com)
- Laghos summary for CTS2 benchmark (Technical Report) | OSTI.GOV
- Nonconforming Mesh Refinement for High-Order Finite Elements | SIAM Journal on Scientific Computing
- https://mfem.org/
- https://ceed.exascaleproject.org/
- https://visit-dav.github.io/visit-website/