A GEOMETRY KERNEL FOR HPC APPLICATIONS

JOHN F. DANNENHOFFER, III* AND ROBERT HAIMES†

∗ Aerospace Computational Methods Laboratory
Syracuse University, Syracuse, NY, 13244
e-mail: jfdannen@syr.edu

†Aerospace Computational Design Laboratory
Massachusetts Institute of Technology, Cambridge MA, 02139
e-mail: haimes@mit.edu

Key words: CFD methodology, Parallel methodology, Parallel CFD applications

Abstract. NASA recently commissioned the study: “CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences”,[1] in which several technology gaps and impediments were identified. The study projected that in the near future, CFD grids with 10^9 nodes (or more) would be commonplace, executing in High Performance Computing (HPC) environments with 10^3 to 10^5 nodes, each consisting of 10s of cores; graphic processing units (GPU) were also seen as technologies that might seriously change the HPC from its current state. Further, the study identified (in several places) the need for widespread adoption of grid adaptation and a transition from one-off analyses to design optimization.

Described herein is a new geometry kernel, EGADSlite, that was developed specifically for HPC environments in order to support parallel mesh generation, solver-based grid adaptation, and the curving of linear meshes to support high(er) order spatial discretizations. The design of the new open-source kernel, which is built upon EGADS[2], is described. In addition, several different applications are detailed. Finally, its use in the Engineering Sketch Pad (ESP)[3, 4] is also described.

In preparation for use the an HPC environment, a specially instrumented version of FUN3D flow solver[5] was used to acquire information about FUN3D’s partitions. An analysis was executed on the AIAA HiLPW-3 JSM test case[6] (see figure) to determine the communication patterns that PAGODA would be expected to support. The data in this table shows that only a very small portion of the configuration is required by each client and that using a bounding box to select portion required 3–10 times as many Faces to be distributed. Even though the List-of-Faces approach is preferred, its implementation requires that each geometry clients knows the identity of its needed Faces.

<table>
<thead>
<tr>
<th>Number partitions</th>
<th>1</th>
<th>10</th>
<th>100</th>
<th>1000</th>
<th>10000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partition w/o Points</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>164</td>
<td>5588</td>
</tr>
<tr>
<td>Avg Faces/Partition</td>
<td>413</td>
<td>54</td>
<td>10</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Avg Faces/bbox</td>
<td>413</td>
<td>140</td>
<td>30</td>
<td>11</td>
<td>3</td>
</tr>
<tr>
<td>Max Faces/Partition</td>
<td>413</td>
<td>102</td>
<td>31</td>
<td>19</td>
<td>13</td>
</tr>
<tr>
<td>Max Faces/bbox</td>
<td>413</td>
<td>413</td>
<td>413</td>
<td>413</td>
<td>413</td>
</tr>
</tbody>
</table>
REFERENCES


