

1. Intro to DPG
 - 1.1. Optimal test functions
 - 1.2. Hybridized DG formulation
2. Implementation details
 - 2.1. Zoltan partitioning
 - 2.2. Everything else (Nate?)
3. Timing runs
 - 3.1. Setup
 - 3.1.1. Mesh
 - 3.1.1.1. Adaptive setup - 2 uniform refinements, 7 refinements along N/E edges.
 - 3.1.1.1. L2 order = 3 (cubic approximations)
 - 3.1.1. Strong scaling
 - 3.1.1.1. Powers of 4 processors, up to 64 (256?)
 - 3.1.2. Weak scaling
 - 3.1.2.1. Powers of 2 processors, up to 128 (8 runs)
 - 3.1.2.1.
 - 3.1. Scaling tests
 - 3.1.1. Stats of interest
 - 3.1.1. Optimal test function computation
 - 3.1.2. Global assembly
 - 3.1.1. Total wall-clock time
 - 3.1.2. Method of scaling
 - 3.1.2.1. Pre-refined mesh for a boundary layer
 - 3.1.2.1. Run initial uniform refinements to increase problem size
 - 3.1.2.1.1. numInitialRefinements = {0,1,2,3}
 - 3.1.2.2. Do 7 refinements on top of these along N/E boundaries
 - 3.1.1. Naive vs guided partitioning
 - 3.1.1. Naive partitioning
 - 3.1.1.1. Use cyclic/random (if i can load-balance random)
 - 3.1.2. Guided partitioning
 - 3.1.2.1. Use HSFC, explain why not using REFTREE