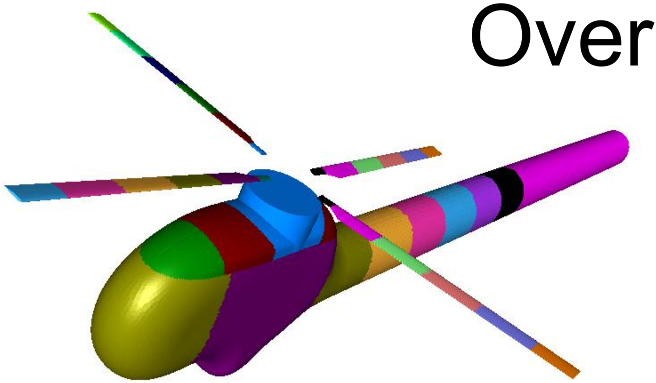
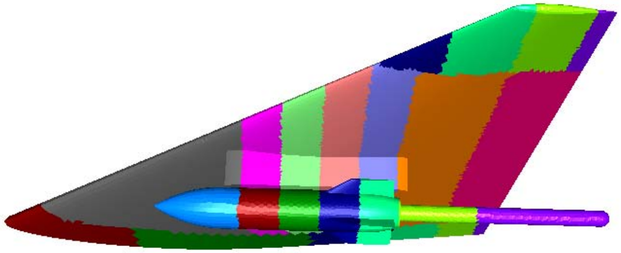
Domain Decomposition for Overset Grid Assembly





Ralph Noack

David Boger

Penn State University Applied Research Laboratory

Outline

- Motivation
- Parallel Decomposition
  - Overset Work
  - New partitioning approach
  - Impact of Partition Boundaries on Overset Work
- Data migration
- Summary

Motivation

- Overset approach
  - Simplify grid generation for complex geometry
  - Enable moving body simulation
- Must compute overset domain connectivity information at each time step
  - Can be time consuming
    - Flow solver scales better than overset computation
- Parallel execution required to reduce wall clock
  - Scaling requires partitioning

Parallel Decomposition

- Parallel execution requires efficient distribution of work across available resources

- Better performance is achieved by minimizing communication between processes

- Scalable parallel execution requires
  - Distribution of work without increasing work
  - Low communication overhead

Flow Solver
Parallel Decomposition

- Work
  - Proportional to number of grid elements
  - Balanced by putting equal numbers of elements on each parallel processor
  - No extra work because of partition boundary (just communication)

- Communication
  - Comprised of data exchanged between neighboring elements
  - Proportional to the number of grid element faces on the boundary between grid partitions
  - Minimized by minimizing number of faces on partition boundaries

- Decomposition
  - METIS software is typically used to partition the grid

Overset Connectivity and Parallel Decomposition

- Hole-Cutting

  Identify locations inside geometry and behind symmetry planes
  - Work
    - Related to the surface area of cutting geometry (or volume inside the geometry)
    - Grids that do not overlap geometry are not cut (no hole-cut work)
  - Communication
    - Minimized by duplicating hole cut geometry

- Donor Search

  Find interpolation source for fringe points
  - Work
    - Related to the number of elements in the fringe grid in the overlap region
  - Communication
    - Minimized by keeping fringe and donor on same rank

- Both are spatial connections
  - Given x,y,z find containing geometry or donor

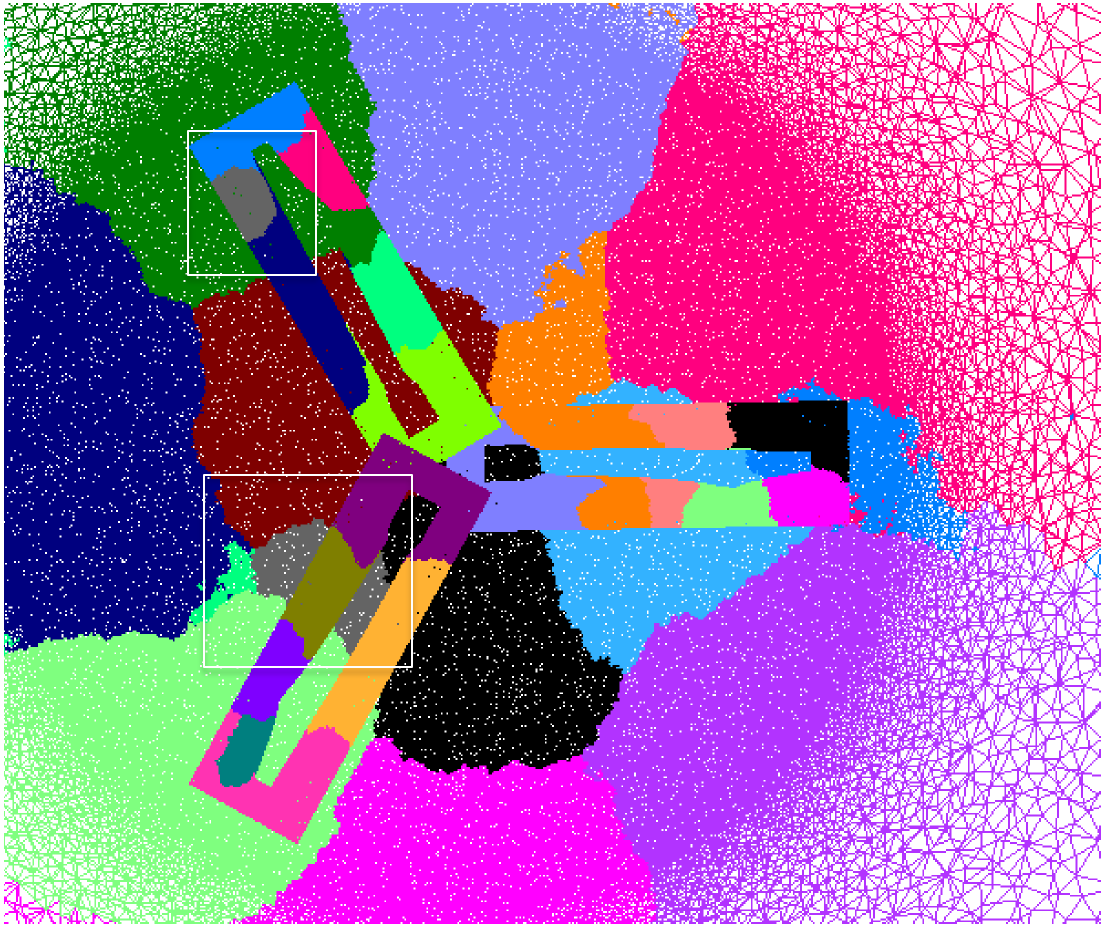
Overset Connectivity Spatially Connected

- Work
  - **Flow solver** work is over the entire domain
  - **Overset domain connectivity** work is only in regions of overlapping grids
    - Large portion of domain is inactive

- Communication
  - Flow solver data exchange is along neighbor connections
  - Overset connectivity data exchange is along spatial connections
  - Using the flow solver decomposition has high probability of maximizing communications
    - Overlapping grids are assigned to different ranks
  - Overset communication: need a **spatial decomposition**

Flow Solver Decomposition

- Cutplane with partitioned grids colored by rank
- 32 processors
- Only the two gray grid partitions are on the same rank, but they do not overlap

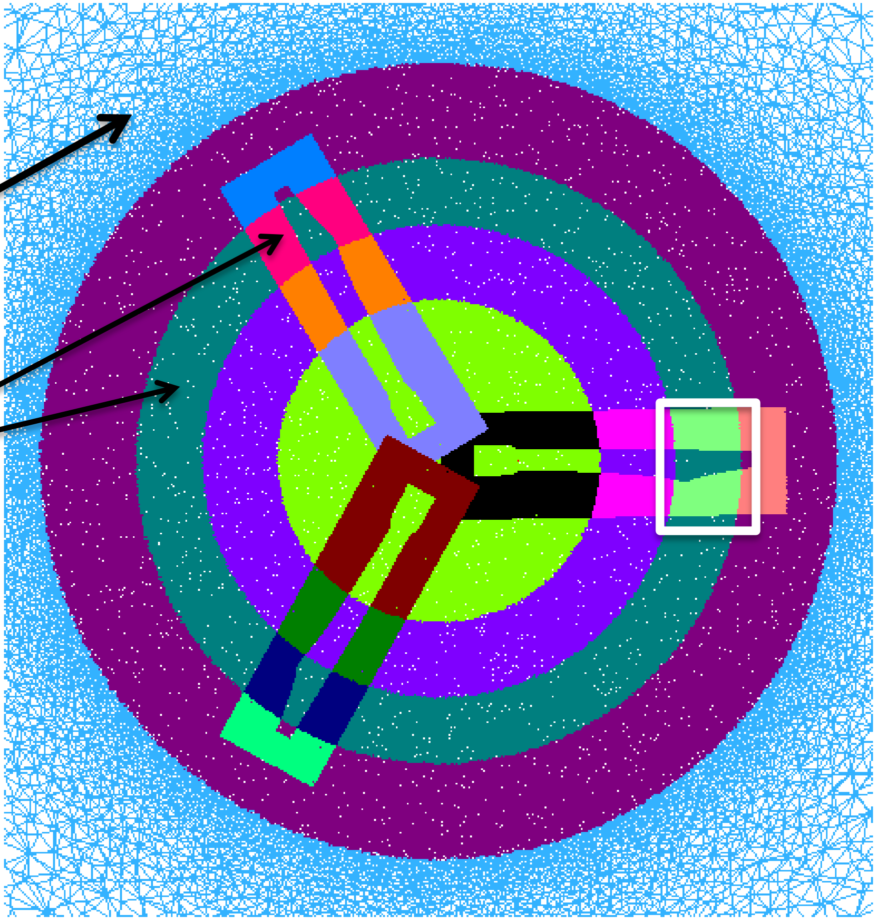


New Decomposition Approach

- Suggar++ uses a new spatial decomposition approach
  - A specified volume is used to assign rank
  - Elements that overlap the same volume are assigned to the same rank (regardless of grid)
  - Currently have two spatial decomposition volume (SDV) types
    - Cartesian box
    - Cylinder
  - Also have ParMETIS for flow-solver-type decomposition

Spatial Decomposition Volume Cylinder

- Cylindrical slices assigned to different ranks
- Outer (Cyan) portion of background grid is inactive
- Elements overlapping cylinder are assigned to rank
- Slice of (rigid) blade will always overlap slice of background grid
- Fringes & donors on same rank



Spatial Decomposition For Store Separation

- Store separation is not constrained to a cyclic region
  - Cylindrical spatial decomposition is inappropriate
  - Cartesian Spatial Decomposition Volume can be used
    - Bounding box of store grid
    - Volume outside the bounding box is inactive
- Will work well in minimizing communications for static problems
- Data migration needed for moving body

Partitioning Can Increase Overset Work

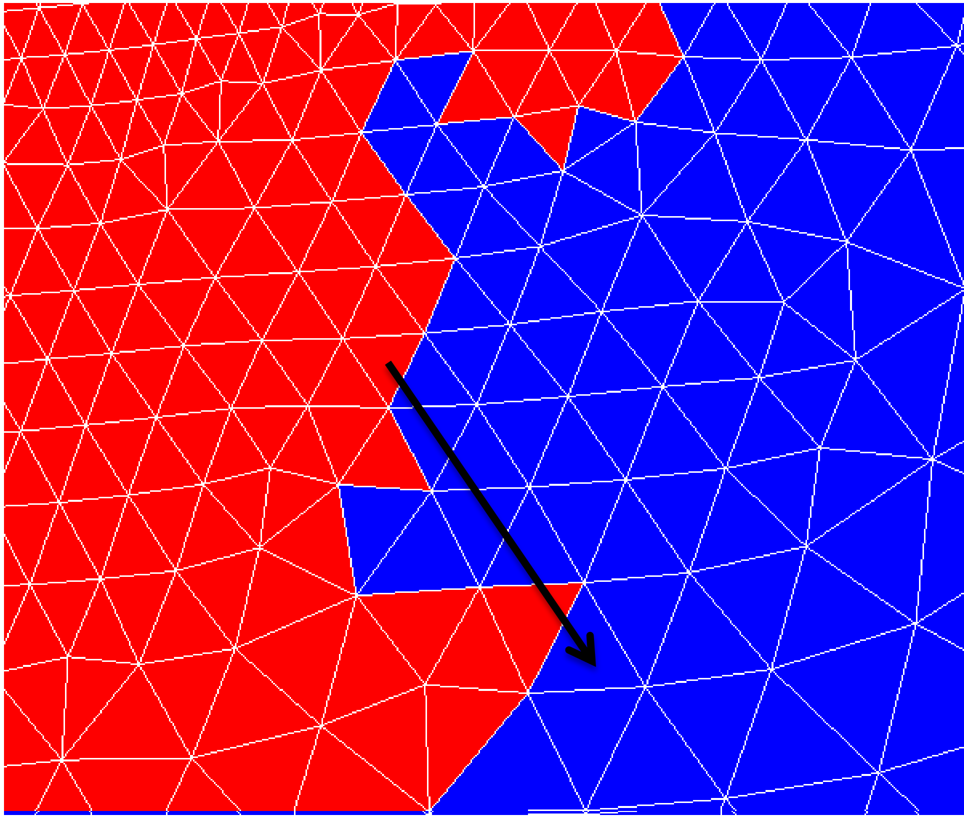
- Parallel partitioning introduces a partition boundary
  - Flow solver: does not change the work
    - Linear scalability
  - Overset grid assembly: can increase the work!
    - Limits scalability

Overset Donor Search
No Partitions

- Start donor search from any location

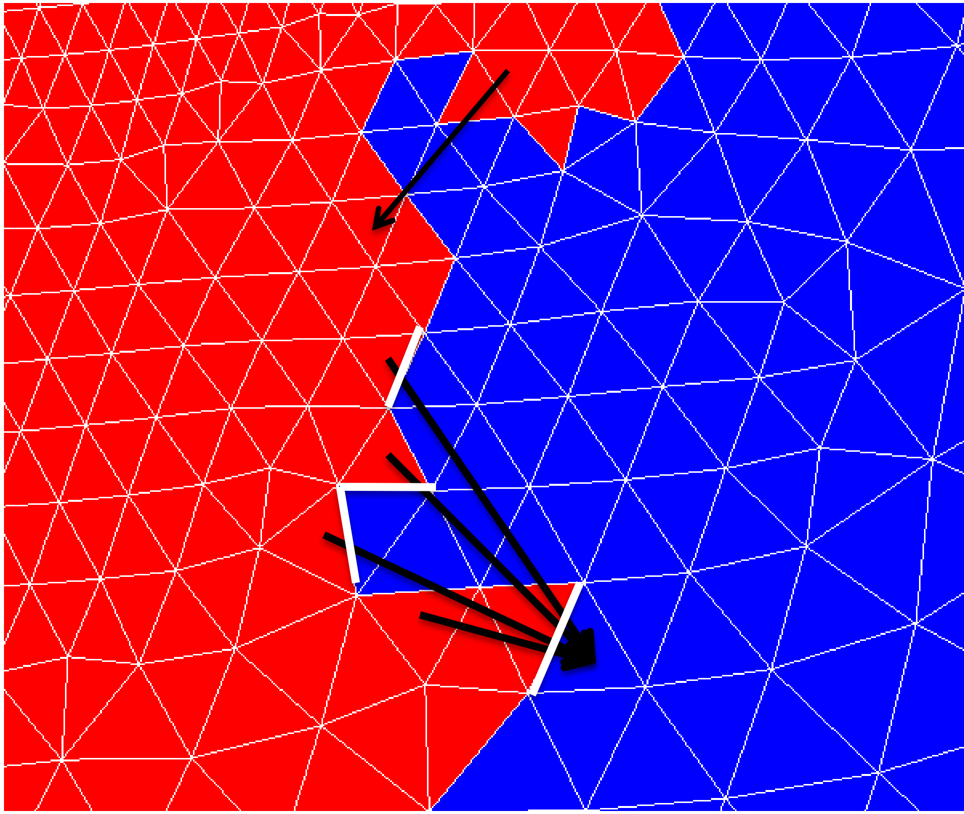
- Will find donor if not crossing a grid boundary

- Parallel partitioning introduces a partition boundary



Overset Donor Search With Partitions

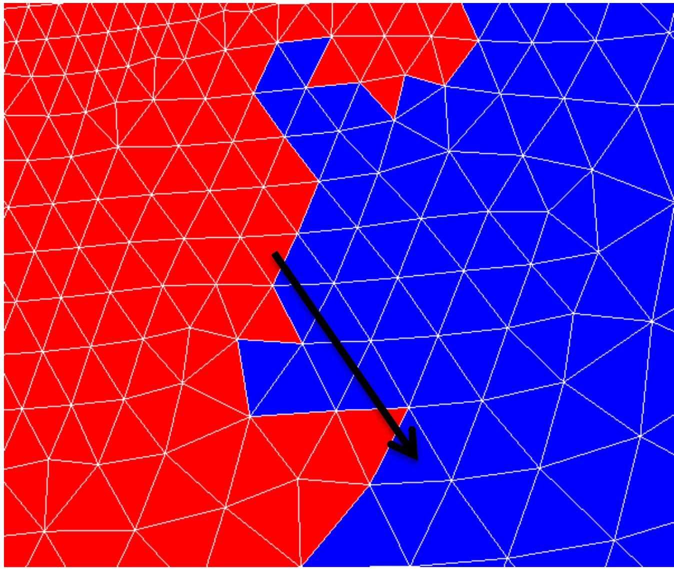
- Start donor search from same location
- Search dead ends at boundary
- Restart from other boundary elements
- Exhaust possible starts: is not in the red grid
- Fringe & Donor in same grid: same problem



Partition boundary requires more donor searches!

Overset Donor Search With Partitions

- Partition boundary requires more donor searches!
  - Work increases with number of partitions
  - Limits scalability
- Ways to reduce searches
  - Suggar++ uses a Boundary Element Alternating Digital Tree to find candidate starts elements
  - Beggar uses a Binary Space Partition Tree to determine if point is inside grid
  - All still require extra work



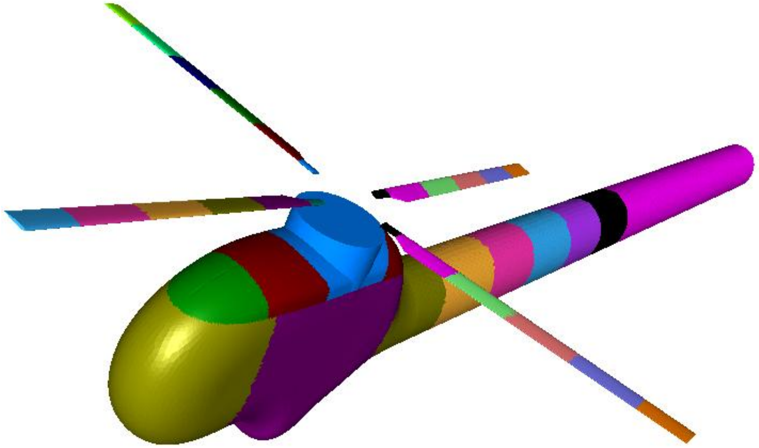
How To Eliminate Extra Searches

- To eliminate extra searches
  1. Donor search for fringe must search in a single partition
  2. Must find ALL possible donors

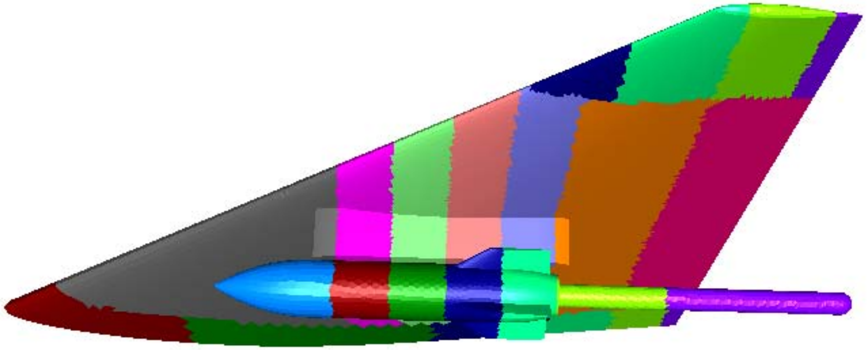
- SDV partitioning provides mechanism
  1. Assign fringe to rank based upon SDV
  2. Rank must contain all elements that overlap associated SDV
     - Elements are assigned to a unique rank/subgrid
     - Include fragments of other subgrids on a rank

Test Cases

- HART II grid
  - 4 blades + fuselage
  - 13.6 million points
  - 79.7 million elements
  - Node-centered assembly

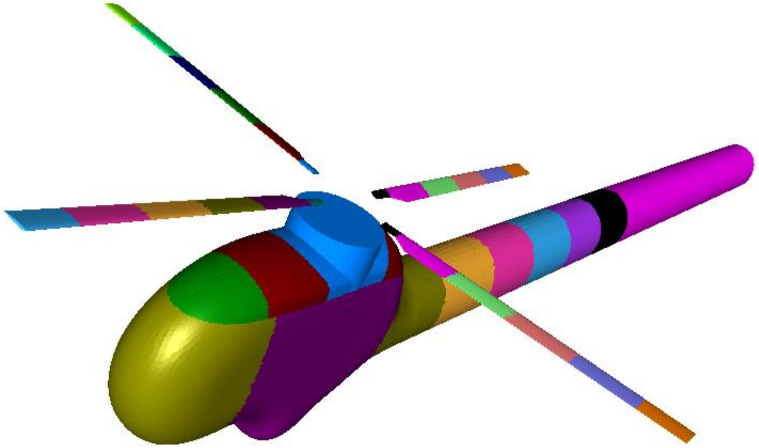


- Eglin Wing/Pylon/Store
  - Store + Wing&Pylon grid
  - 1.3 million points
  - 7.5 million elements.
  - Cell-centered Assembly



ParMETIS vs Spatial Decomposition HART II

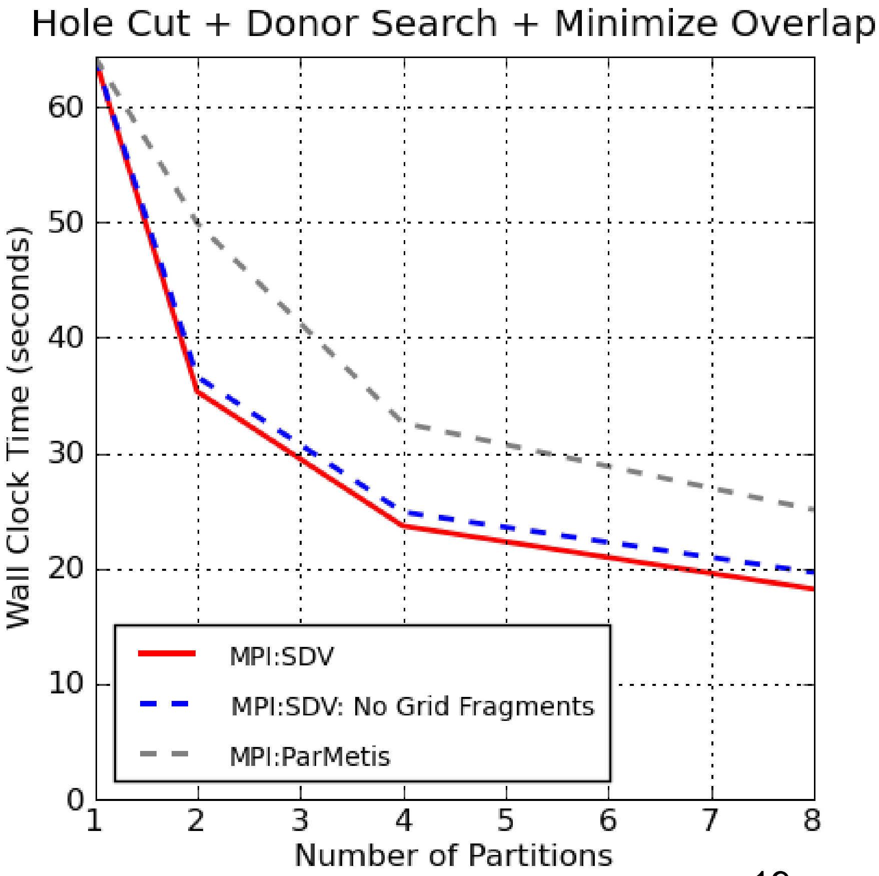
- Decomposition
  - 8 processors
  - ParMETIS
    - Flow solver type decomposition
  - Cylindrical SDV



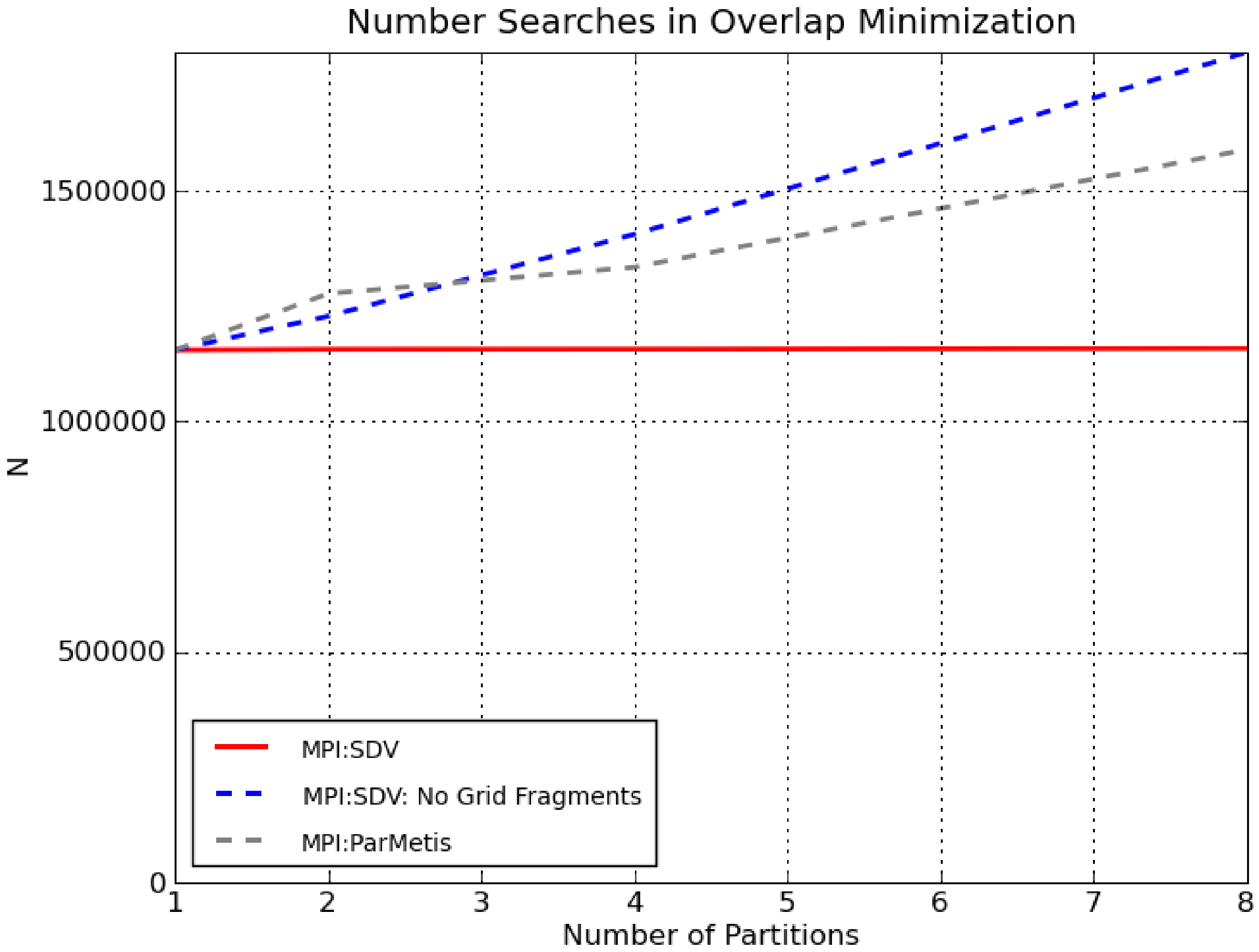
- ParMETIS decomposition
  - NO donor searches on the same rank
- Spatial decomposition
  - ALL donor searches on the same rank

ParMETIS vs Spatial Decomposition HART II

- Compare work for a time step
  - Does not include I/O
- SDV: No Fragments and ParMETIS
  - Include extra searches due to partitioning
  - Differences in load balance
  - SDV has more searches on rank (less communication)
- SDV improves performance

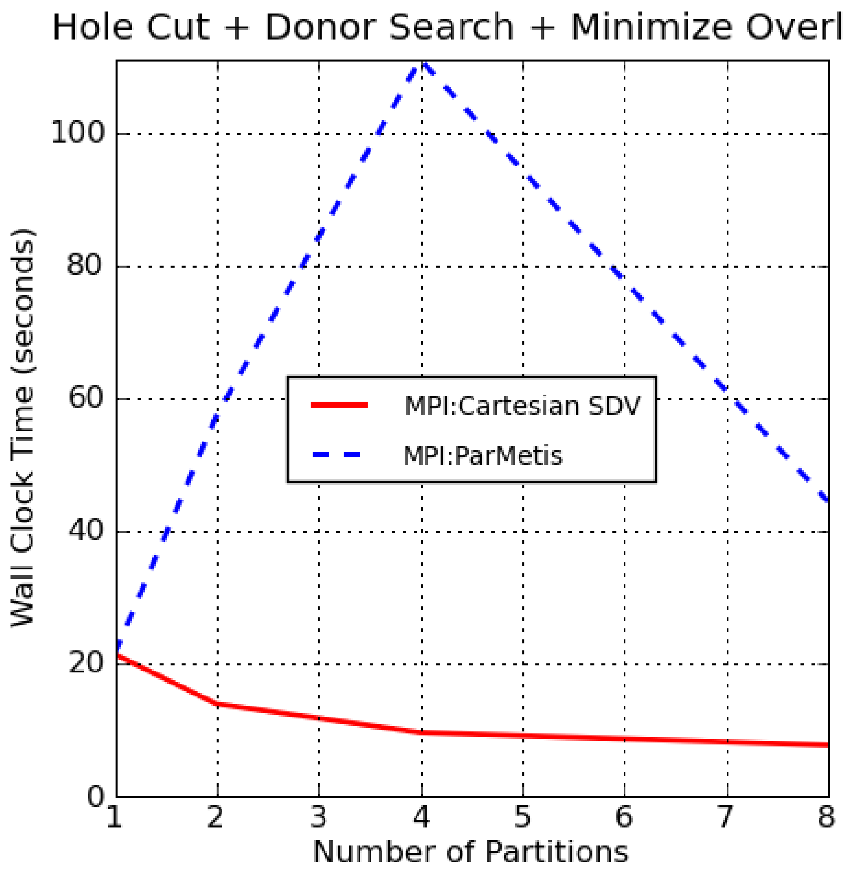


ParMETIS vs Spatial Decomposition HART II

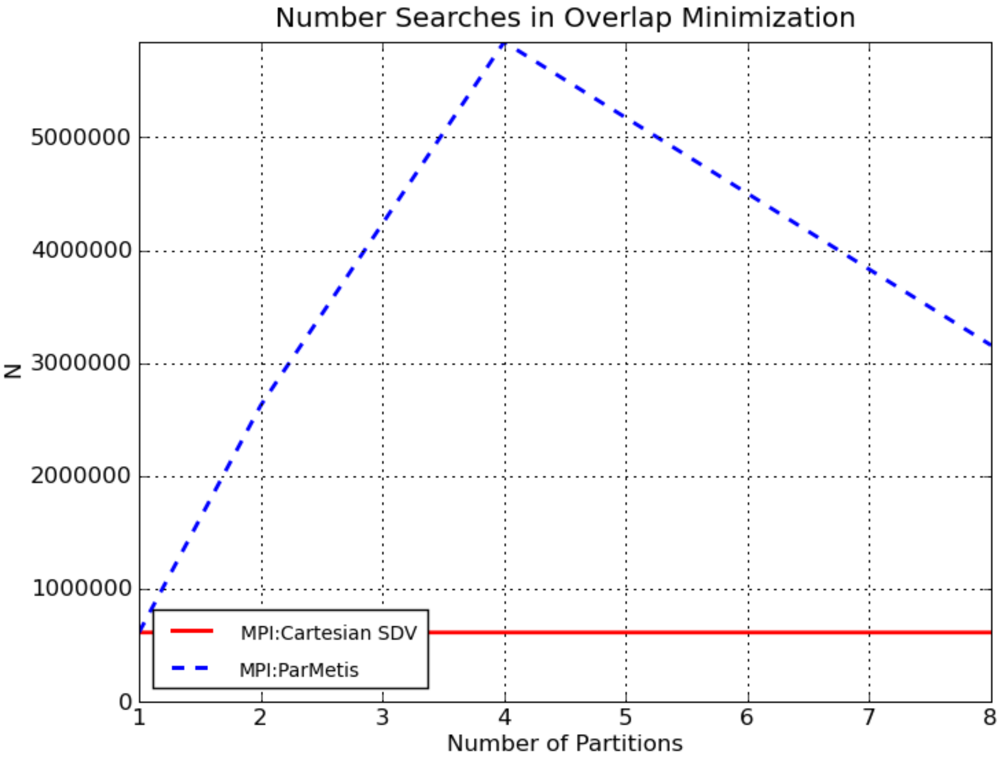


PENNSTATE ARL

ParMETIS vs Spatial Decomposition
Eglin Wing/Pylon/Store

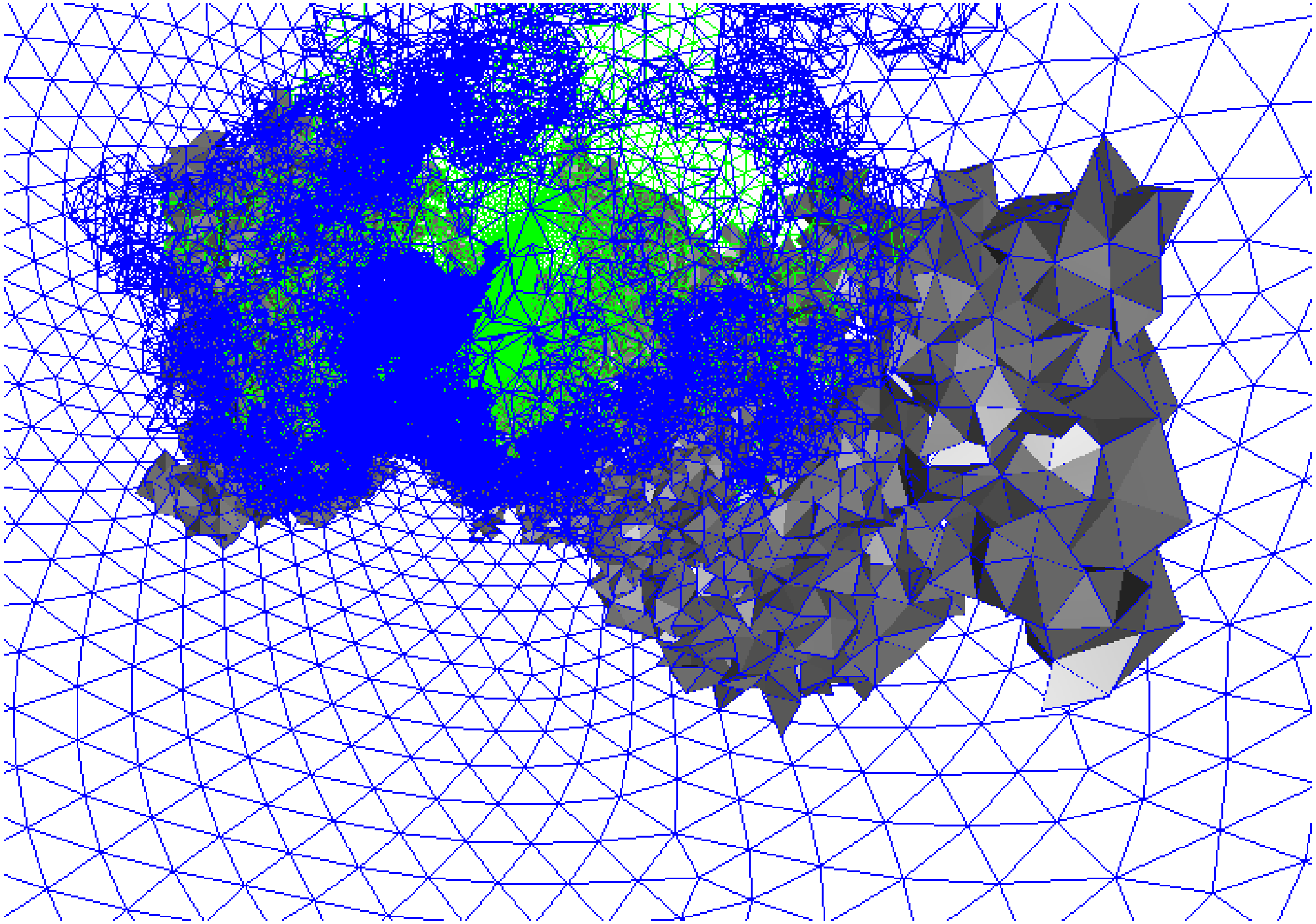


Hole Cut + Donor Search + Minimize Overlap



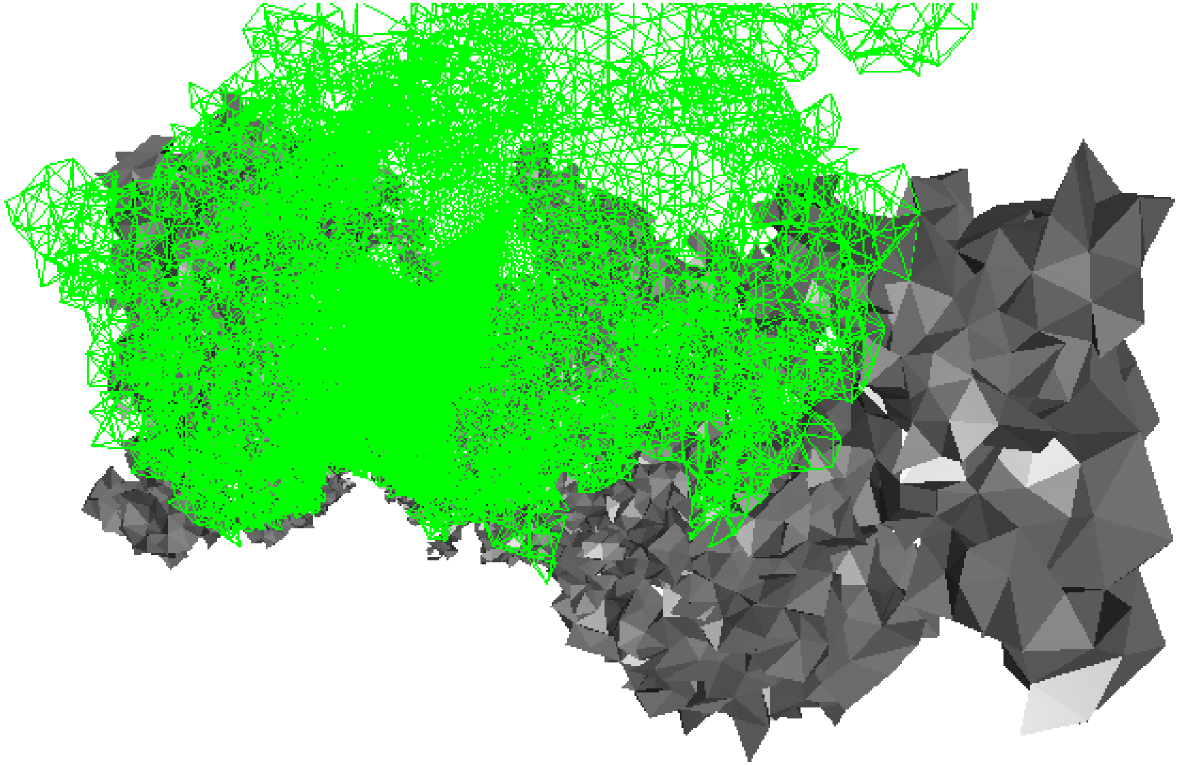
Number Searches in Overlap Minimization

ParMETIS Partitions: NP=4
Eglin Wing/Pylon/Store



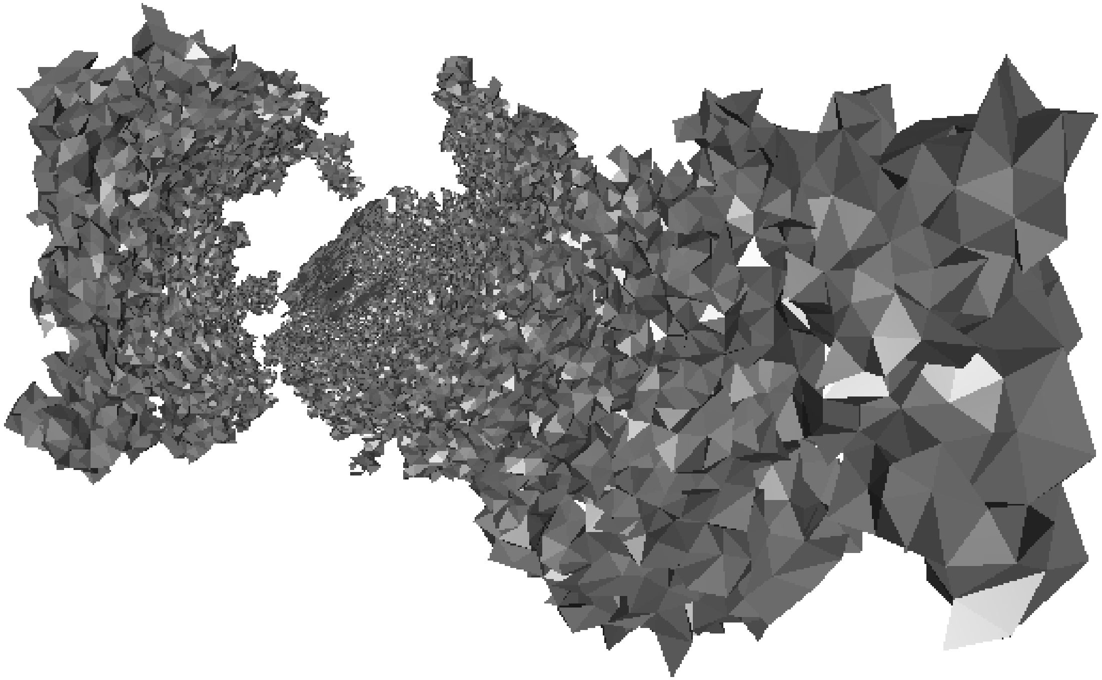
Wing/Pylon Grid: Partitions 0,1,2

ParMETIS Partitions: NP=4
Eglin Wing/Pylon/Store



Wing/Pylon Grid: Partitions 1,2

ParMETIS Partitions: NP=4
Eglin Wing/Pylon/Store



This partitioning is not well suited to overset assembly

Wing/Pylon Grid: Partition 1

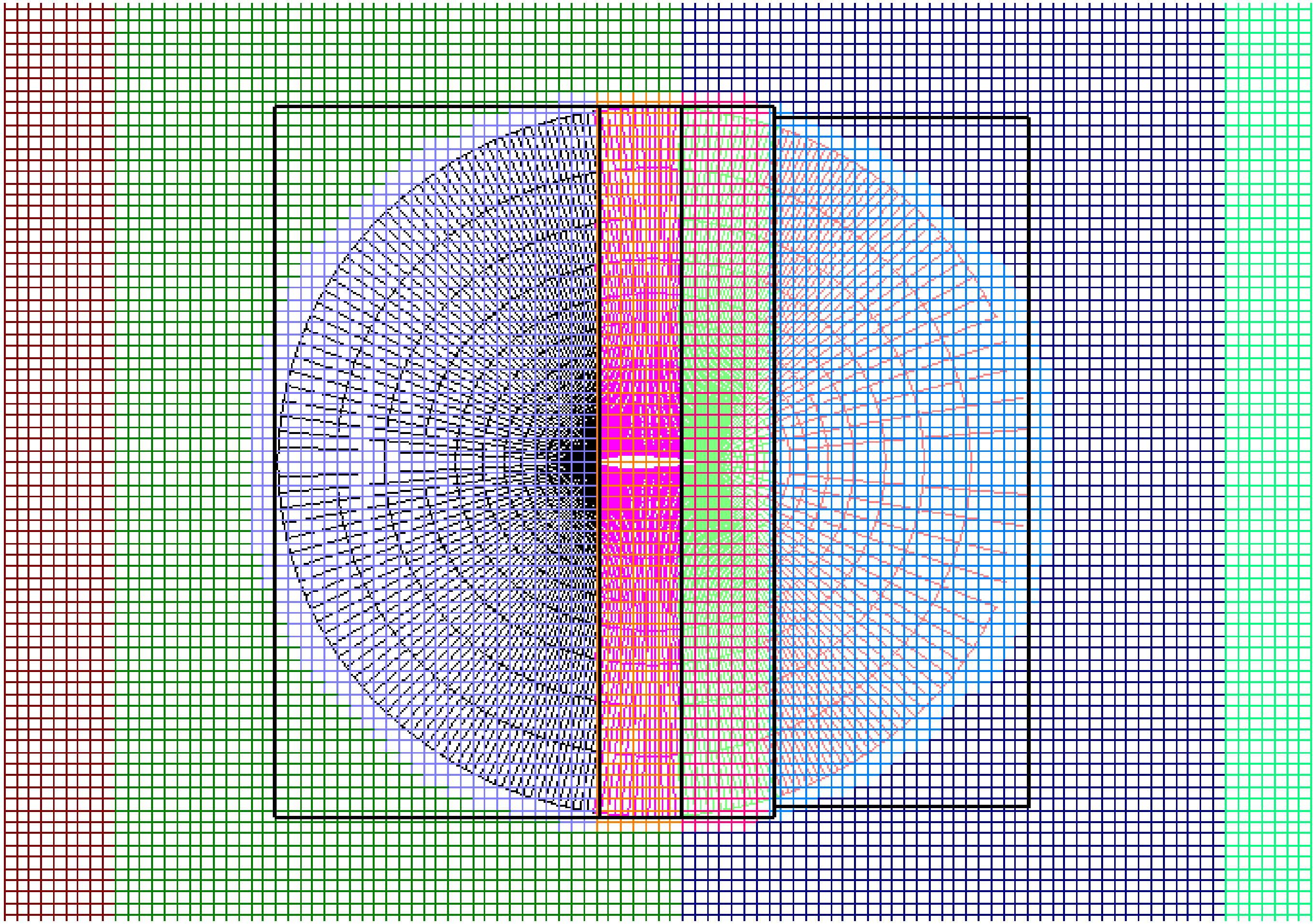
Data Migration

- Migration of grid data between ranks is required:
  - – Cartesian SDV with moving bodies
  - – Cylindrical SDV with non-rigid motion
    - Lead/Lag/Flap with rigid blade
    - Deforming blade
  - – Load balancing
- Requires lots of work to modify grids in each partition

Data Migration Status

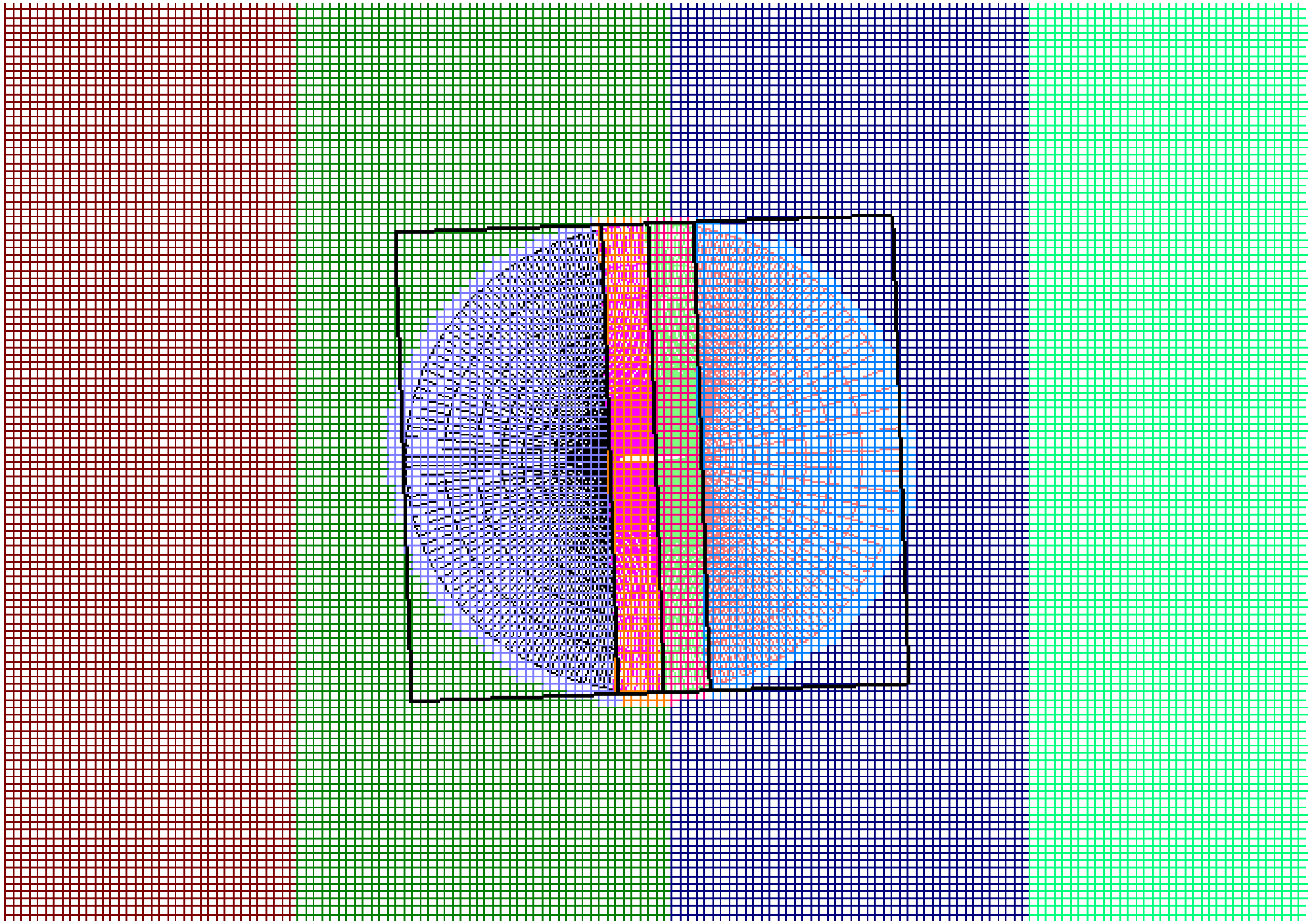
- Working for simple test case
  - Very preliminary results
- Need algorithm for dynamic load balancing
  - How to accurately measure work on rank
- Would like to have data migration/load balance run as a background process

Cartesian SDV with Translation



“Wake” is a result of not “load balancing inactive region

Cartesian SDV with Rotation



“Wake” is a result of not “load balancing inactive region

Summary

- Work in computing overset domain connectivity information is significantly different than flow solver work
  - Work only in portion of domain
  - Communication is along spatial connections

- Partitioning can negatively impact overset Work

Summary(2)

- Partitioning using new Spatial Decomposition Volume can significantly improve parallel performance for overset grid assembly
  - Reduce communication
  - Eliminate extra donor searches/work resulting from partition boundaries

- Data migration required for general use of SDV
  - Preliminary implementation within Suggar++

Summary(3)

- Future work
  - Finish data migration
  - Dynamic load balancing by moving SDV boundaries
  - Internal grid adaptation

Acknowledgments

- Funding provided by
  - National Aeronautics and Space Administration under Agreement No. NNX07AU75A
  - New 6DOF Environment project through the DoD HPC Institute for HPC Applications to Air Armament.