Building parallel information from distributed meshes

March 19, 2012

Abstract

This documents shows how to build parallel information from distributed submeshes without any additional information except vertex coordinates.

MSTK_FunctionName() indicates a serial function without communications. PMSTK_FunctionName() indicates a collective parallel function.

1 Assign global IDs to distributed meshes

Here we are given submeshes on each processor, each submesh is a regular MSTK mesh without any additional information. The aim is to assign each vertex a unique global ID. For a vertex shared by 2 or more processors, we assign its ownership to the lowest rank processor.

PMSTK_AssignGlobalIDs(Mesh_ptr mesh, int rank, int num, MPI_Comm comm)

```
1.  extract boundary vertices list l with vertex coordinates
2.  sort the list based on vertex coordinates values
3.  Allgather number of total and boundary vertices
4.  for i = 0:num-1
5.      Broadcast l with vertex coordinates
6.  for j = 0:rank-1
7.      compare each vertex v on l with vertices from processor j
8.        if there is a match, set the master partition id of v
9.  calculate number of ghost vertices
10. Allgather number of ghost vertices
11. Assign global IDs of owned vertices
11. for i = 0:num-1
12.      Broadcast l with global IDs
13. Assign global IDs of ghost vertices
```

An alternative way to do this is using space-filling curve, but that is generally used for all the vertices on all processors, here we only need to deal with boundary vertices on each processors since we assume there is no overlapping elements.

2 Build connection information across processors

Use vertices global ID to identify neighboring processor information, each mesh object stores a list of neighboring processor IDs.

PMSTK_BuildNborProcs(Mesh_ptr mesh, int rank, int num, MPI_Comm comm) This function builds neighboring processors information. The difference between this function and MESH_Update_ParallelAdj() is that the latter function assumes each submesh has ghost entities, while here we only assume the global IDs are given. This function is similar to PMSTK_AssignGlobalIDs() except it uses global ID as the key instead of coordinate value, also it sets mesh_par_adj_flags and mesh_par_adj_info, similar to function MESH_Update_ParallelAdj()

3 Add ghost cells through communications

PMSTK_BuildGhostElements(Mesh_ptr mesh, int rank, int num, MPI_Comm comm) This function adds ghost elements on each processor.

- MSTK_SendGhostElements(Mesh_ptr mesh, int rank, int num, MPI_Comm comm)

send the list of boundary elements to the neighboring processors.

```
1.  for i = 0:num-1
2.  if processor i has ghost elements on processor rank
3.     send the list of boundary elements to processor i
```

- MSTK_RecvGhostElements(Mesh_ptr mesh, int rank, int num, MPI_Comm comm)

receive the list of boundary elements from the neighboring processors.

```
1.  for i = 0:num-1
2.  if processor rank has ghost elements on processor i
3.     receive the list of boundary elements from processor i
4.  for each element e in the receive buffer
5.      if e has a vertex with the same global ID as a vertex v on processor rank
6.          add e to the neighboring list of v
```