Multi-block meshes are widely used in many simulations. They are defined using a block data structure and an array of nodes within each block. Thus, once a block decomposition is computed, different meshes with different properties can be generated. However, we have to define an efficient way to navigate through the nodes of the mesh. For this reason, we define the multi-block structured iterator to traverse the nodes of these kind of meshes. In addition, we apply the multi-block iterator to implement a smoothing algorithm based on differential operators.

Our Objectives

1. Define a multi-block iterator to navigate through the nodes of a multi-block structured mesh.
2. Implement a smoother algorithm using the multi-block iterator.

Multi-Block Structured Mesh

1. Large blocks define the coarse structure.
2. An array of nodes is defined in each block.

Key Ingredients

1. Low number of blocks compared to number of nodes.
2. Full adjacency information is stored for the blocks.
3. Mesh adjacencies are computed using the block structure.
4. Local mappings transform the logical coordinates of adjacent blocks.

Multi-Block Iterator

1. Define initial block and the range of nodes to traverse.
2. Blocks to traverse are stored in a heap.
3. End signal is emitted when the heap is empty.

Examples

- Draft Tube
  - Winslow operator
- Draft Tube
  - Original Mesh
- Draft Tube
  - Smoothed Mesh

Ref.