May 10, 2022

Ok so here are a few ways to go about this

1. redo Mesh and Field in Morpho
2. create a sublcass of Mesh and Field (the GPU version)
3. have Mesh and Field store both CPU and GPU data structres and switch off between them

Considerations on GPU data structures

- The Matrix class now has an equivalent GPU implementation (fully for CUDA and everything but div for openCL)
- The Sparse matrix here is only used as a data structure so the GPU implementation only needs minimal interface

1 Mesh

What follows here is a bit of design explaniation and function description for the current (0.5.3) implemenation of Mesh. The overview is that a mesh of dimention D and N vertices is described by a $D \times N$ matrix of vertex positions. Additionally the mesh has a $(D+1)\times(D+1)$ Array of Sparse Matrices describing the conncectivty between these vertices.

The public interface to Mesh is

- print()
- save(string filename)
- vertexmatrix()
- setvertexmatrix(Matrix vert)
- vertexposition(int i)
- setvertexposition(int i, Matrix position)
- resetconnectivity()
- connectivitymatrix()

- addgrade(int grade)
- maxgrade()
- addsymmetry()
- transform()
- clone()
- count()

Items in pink are generic morpho methods, print (save here write the mesh to a file). Blue are fairly straight forward getters and setters. **Of note here is the use of get column and set column for vertex position** The items left in black deal are a bit more complicated are each gets their own subsection below

1.1 connectivitymatrix(int row, int column

) This is the getter for the spare connectivity matrices. Its more complicated than one might think because connectivity matrices are lazily created so requesting one might lead to its creation form equivalent information

1.2 addgrade(int grade, (optional) Sparse Connectivity)

This function adds a grade to a mesh. Its actually performs two different operations depending on the optional sparse argument.

If the sparse argument is given it acts as a simple setter, calling mesh_setconnectivityelement to delink the old mesh, link the new one and put it in the connectivity array. The mesh's connectivy matrices for each grade are then flipped to ccs mode (if they weren't already).

If add grade is called with just an intger addgrade will populate lower connectivity elements from top ones unill it reaches the requested grade (use volumes to generate facets, facets to generate lines). this is done though a call to mesh_addgrade(mesh m, int grade)

1.2.1 mesh_addgrade(mesh m, int grade)

This function does the following

check if the grade (item [0,grade] in the array) already exists, if so return it

Find the smallest grade larger than the requested one.