

*Compte rendu : Intégration de LBCPL dans LibMultiScale  
Adaptater les structures de données et nouvelles fonctions.*



Equilibrage de charge et distribution de données pour  
des simulations parallèles irrégulières couplées :  
application à la propagation multi-échelle de fissures  
dans les matériaux.

Maîtres de stage :  
Olivier COULAUD  
Aurélien ESNARD  
Jean ROMAN

Encadrant pédagogique :  
François PELLEGRINI

Période de stage :  
du 09 février 2009 au 30 septembre 2009

Mohamed Amine EL AFRIT  
[www.melafrit.com](http://www.melafrit.com)  
postmaster@melafrit.com

# Intégration de *LBCPL* dans *LibMultiScale*

*LBCPL* est le nom de notre module d'équilibrage de la charge : *Load Balance Coupling Codes Module*.

Un patch contenant tous ces modifications est disponible dans le répertoire **lbcpl/patchLibMultiScale-Simulpack** du svn à l'adresse <http://gforge.inria.fr/projects/mpicpl/>.

## Patch SimulPack

*SimulPack* est le code de la mécanique des milieux continus utilisé (dans notre cas) pour la simulation macroscopique.

## Adaptation avec les structures de données dde SimulPack

Dans *SimulPack*, il y a des variables globales qui sont utilisée et qu'il faudrait remplir au chargement du maillage. Comme nous utilisons une structure de donnée différente il faudrait adapter.

Les variables globales sont décrites dans le fichier **simulpack/mechanics/data.h**. Les variables qui nous intéressent (on décrit en commentaire à quoi chaque variable correspond dans notre structure de donnée) sont :

```
1 extern double * coordinates; /* nodal coordinates mesh->nodes */
2 extern int * connectivity; /* element connectivity */
3 extern int * element_type; /* element type depends on mesh->elm_size */
4 extern int nodes; /* current number of nodes mesh->nb_nodes */
5 extern int nodes_element; /* number of nodes/element mesh->elm_size */
6 extern int elements; /* current number of elmts mesh->nb_cells */
7 extern int dof_node; /* number of dof per node */
8 extern int spatial_dimension; /* spatial dimension */
```

Il faudra ajouter **struct.h** (qui décrit notre structure de donnée) dans **simulpack/**. Ce fichier se trouve dans le répertoire **lbcpl/src**.

Les allocations mémoires doivent se faire de la manière suivante :

```
1 coordinates = malloc(max_node_label*spatial_dimension*sizeof(double));
2 connectivity = malloc(elements*nodes_element*sizeof(int));
3 element_type = malloc(elements*sizeof(int));
```

Les fichier cmake à modifié pour la compilation avec CMake sont :

- **simulpack/CMakeLists.txt**;
- **simulpack/mesh\_io/CMakeLists.txt**;

Dans **simulpack/mesh\_io/mesh\_io.h**, ajouter la declaration :

```
1 ...
2 extern void load_mesh_MSH(const char * inputfile);
3 ...
```

Ajouter la fonction `void load_mesh_MSH(const char * inputfile)` dans **simulpack/mesh\_io/unv\_io.c**

```
1 void load_mesh_MSH(const char * inputfile){
2
3     /* load mesh */
4     Mesh mesh;
```



```

13         coupling_he_cost);
14
15     int ubfactor_brhgh = UBFACITOR;
16     int part_brhgh[brhgh.nvtxs];
17     int edgecut_brhgh = 0;
18
19     partitionHypergraph(&brhgh, nparts, ubfactor_brhgh,
20                       part_brhgh, &edgecut_brhgh);
21
22     /* partitioning with fix cells */
23     int ubfactor_fixhg = UBFACITOR;
24     int part_fixhg[hg.nvtxs];
25     int edgecut_fixhg = 0;
26     // free cells
27     for(int i = 0 ; i < hg.nvtxs ; i++) part_fixhg[i] = -1;
28     // pre-assigned cells
29     for(int i = 0 ; i < brhgh.nvtxs ; i++) part_fixhg[i] = part_brhgh[i];
30
31     partitionHypergraphWithFixCells(&hg, nparts, ubfactor_fixhg,
32                                   part_fixhg, &edgecut_fixhg);
33
34     /* add hypergraph partitioning variable */
35     double parttmp_fixhg[hg.nvtxs];
36     for(int i = 0 ; i < hg.nvtxs ; i++) parttmp_fixhg[i] = part_fixhg[i];
37
38     addMeshCellVariable(&mesh, CELL_FIELD_FIXHGPART_ID, 1, parttmp_fixhg);
39
40     /*use parttab for the following functions*/
41     partab = parttmp_fixhg;
42     fprintf(stderr, "(%d) partitioning of mesh in %d pieces: done\n",
43            adlib_myrank, adlib_world_size);
44 }

```

Mise à jour du fichier **simulpack/mesh\_io/CMakeLists.txt** pour prendre en compte le nouveau fichier **msh\_io.c** :

```

1  /* simulpack/mesh_io/CMakeLists.txt */
2  SET(libmesh_ioSources
3    unv_io.c
4    msh_io.c
5  )

```

Mise à jour du fichier **simulpack/mesh\_io/CMakeLists.txt** pour prendre en compte le module **LBCPL** :

```

1  /* simulpack/mesh_io/CMakeLists.txt */
2  OPTION(USING_HGGEN "use hggen" OFF)
3  IF (USING_HGGEN)
4  FIND_PATH(HGGEN_INCLUDE_DIR hggen.h
5            /usr/include
6            $ENV{HGGEN_TOP}/include
7            ${HGGEN_DIR}/include
8            ${HGGEN_DIR}/src
9  )
10
11  FIND_LIBRARY(HGGEN_LIBRARY hggen

```

```

12     /usr/lib
13     $ENV{HGGEN_TOP}/lib
14     ${HGGEN_DIR}/lib
15     ${HGGEN_DIR}/src
16 )
17
18 IF (HGGEN_LIBRARY AND HGGEN_INCLUDE_DIR)
19     SET(HGGEN_FOUND TRUE)
20 ELSE(HGGEN_LIBRARY AND HGGEN_INCLUDE_DIR)
21     SET(HGGEN_FOUND FALSE)
22     MESSAGE("HGGEN not found, set HGGEN_DIR.")
23     SET(HGGEN_DIR "" CACHE PATH "Root path to hggen library")
24 ENDIF (HGGEN_LIBRARY AND HGGEN_INCLUDE_DIR)
25 ENDIF (USING_HGGEN)

```

## Patch LibMultiScale

Les fichier cmake à modifié pour la compilation avec CMake sont :

- **libmultiscale/src/amel/CMakeLists.txt** ;
- **libmultiscale/src/cmake/FindHGGEN.cmake**.

Mise à jour du fichier **libmultiscale/src/amel/CMakeLists.txt** pour prendre en compte de **LBCPL** :

```

1 /*libmultiscale/src/amel/CMakeLists.txt*/
2 IF (USING_HGGEN)
3 TARGET_LINK_LIBRARIES(AMEL ${HGGEN_LIBRARY})
4 TARGET_LINK_LIBRARIES(LMPOST ${HGGEN_LIBRARY})
5 ENDIF (USING_HGGEN)

```

Ajouter le fichier **libmultiscale/src/cmake/FindHGGEN.cmake** pour prendre en compte **LBCPL** et qui contient :

```

1 /*libmultiscale/src/cmake/FindHGGEN.cmake*/
2 ...
3 OPTION(USING_HGGEN "use hggen" OFF)
4 IF (USING_HGGEN)
5 FIND_PATH(HGGEN_INCLUDE_DIR hggen.h
6     /usr/include
7     $ENV{HGGEN_TOP}/include
8     ${HGGEN_DIR}/include
9     ${HGGEN_DIR}/src
10 )
11
12 FIND_LIBRARY(HGGEN_LIBRARY hggen
13     /usr/lib
14     $ENV{HGGEN_TOP}/lib
15     ${HGGEN_DIR}/lib
16     ${HGGEN_DIR}/src
17 )
18
19 IF (HGGEN_LIBRARY AND HGGEN_INCLUDE_DIR)
20     SET(HGGEN_FOUND TRUE)
21 ELSE(HGGEN_LIBRARY AND HGGEN_INCLUDE_DIR)
22     SET(HGGEN_FOUND FALSE)

```

```
23     MESSAGE("HGGEN not found, set HGGEN_DIR.")
24     SET(HGGEN_DIR "" CACHE PATH "Root_path_to_hggen_library")
25 ENDIF(HGGEN_LIBRARY AND HGGEN_INCLUDE_DIR)
26 ENDIF (USING_HGGEN)
```

Mise à jour de la fonction : *void simulpack\_init(double ts)*  
dans **libmultiscale/src/continuum/simulpack/simulpack\_cfunctions.c** avec :

```
1  ...
2  int NEW_STRATEGY = 1;
3  int OLD_STRATEGY = 0;
4  int strategy = NEW_STRATEGY;
5  if (strategy == NEW_STRATEGY) {
6      load_mesh_MSH(fname);
7      partition_mesh_hg(adlib_world_size);
8  }
9  else if (strategy == OLD_STRATEGY){
10     load_mesh_UNV(fname);
11     partition_mesh(adlib_world_size);
12 }
13 ...
```

Les variables **NEW\_STRATEGY** et **OLD\_STRATEGY** désignent le mode de partitionnement (en utilisant ou pas le module **LBCPL**). Ces variables sont temporairement utilisées ici et seront détectées automatiquement par le parseur ultérieurement.