

Procédure de rajout et de test du mailleur tétraédrique Netgen dans le module SMESH de l'environnement Salome 2

Préalables:

On suppose que Netgen a été installé dans la configuration suivante :

```
ls netgen_installation_path/include
nlib.h
ls netgen_installation_path/lib/LINUX
libcsg.a libgprim.a libmesh.a libopti.a libvis.a
libgen.a libla.a libnginterface.a libstlgeom.a
ls netgen_installation_path/bin/LINUX
dialog.tcl  menustat.tcl  nghelp.tcl  ng.tcl  parameters.tcl  variables.tcl
drawing.tcl  ng  ngicon.tcl  ngvisual.tcl  startup.tcl
```

où `netgen_installation_path` est la directory d'installation de Netgen. Les fichiers `netgen_installation_path/bin/LINUX/*.tcl` sont les fichiers de command tcl pour piloter le mailleur Netgen à travers son IHM. `netgen_installation_path/bin/LINUX/ng` est l'exécutable de Netgen avec son IHM embarquée. Les librairies *.a dans `netgen_installation_path/lib/LINUX/` doivent être compilés en enlevant l'option `-DOPENGL` qui n'est utile que pour l'IHM de Netgen. La librairie `libnginterface.a` doit contenir l'objet `nlib.o` et ne doit pas contenir `ngnewdelete.o`.

La procédure à adopter est tout d'abord de compiler Netgen (voir le README.INSTALL venant avec la distribution); ce qui produira l'exécutable `ng` lié statiquement avec les librairies *.a. Puis après il faudrait modifier les différents Makefile pour enlever l'option `-DOPENGL`, rajouter l'objet `nlib.o`, exclure l'objet `ngnewdelete.o` à la librairie `libnginterface.a` et enfin recompiler seulement les librairies.

- On a à notre disposition (de la part de Nadir) une archive `Netgen.tgz` contenant les sources de SMESH qui wrappent les appels aux routines de Netgen pour le mailleur tétraédrique:

```
cd SALOME2_ROOT
tar zxvf Netgen.tgz
./SMESH_SRC/src/NETGEN/Makefile.in
./SMESH_SRC/src/SMESH/SMESH_NETGEN_3D.cxx
./SMESH_SRC/src/SMESH/SMESH_NETGEN_3D.hxx
./SMESH_SRC/src/SMESH/SMESH_MaxElementVolume.cxx
./SMESH_SRC/src/SMESH/SMESH_MaxElementVolume.hxx
./SMESH_SRC/src/SMESH_I/SMESH_NETGEN_3D_i.cxx
./SMESH_SRC/src/SMESH_I/SMESH_NETGEN_3D_i.hxx
./SMESH_SRC/src/SMESH_I/SMESH_MaxElementVolume_i.cxx
./SMESH_SRC/src/SMESH_I/SMESH_MaxElementVolume_i.hxx
./SMESH_SRC/src/SMESH_I/SMESH_LengthFromEdges_i.cxx
./SMESH_SRC/src/SMESH_I/SMESH_LengthFromEdges_i.hxx
./SMESH_SRC/src/SMESH_SWIG/SMESH_box_tetra.py
./SMESH_SRC/src/SMESH_SWIG/SMESH_box2_tetra.py
./SMESH_SRC/src/SMESH_SWIG/SMESH_box3_tetra.py
./SMESH_SRC/src/SMESH_SWIG/SMESH_fixation_tetra.py
./SMESH_SRC/src/SMESH_SWIG/SMESH_fixation_hexa.py
./SMESH_SRC/src/SMESH_SWIG/SMESH_Partition1_tetra.py
./SMESH_SRC/src/SMESH_SWIG/SMESH_flight_skin.py
./SMESH_SRC/adm_local/unix/config_files/check_Netgen.m4
```

La directory SALOME2_ROOT est supposée être la directory contenant tous les modules de la plateforme SALOME 2. Les fichiers *MaxElementVolume* sont déjà dans la base mais ne contiennent rien de signifiant.

- Modification du fichier *configure.in.base* pour rajouter la vérification du fichier include nglib.h et des différentes librairies Netgen, via le fichier m4 check_Netgen.m4.

```
echo testing netgen
echo -----
echo
CHECK_NETGEN
echo
echo -----
```

- Modification du fichier *Makefile.in* pour pouvoir utiliser le dessin mesh_tree_algo_tetra.png dans l'IHM de SMESH (voir la modif à faire sur SMESHGUI) et la Brep flight_solid.brep dans le test SMESH_flight_skin.py.

```
RESOURCES_FILES = \
.
.
.
mesh_tree_algo_tetra.png \
flight_solid.brep
```

- Le fichier *adm/unix/make_commence.in* doit aussi être modifié à cette fin :

```
ifeq (@WITHNETGEN@,yes)
ACLOCAL_SRC += check_netgen.m4
endif
```

- Modification du fichier *SMESH_SRC/src/Makefile.in* pour rajouter la directory NETGEN où la librairie dynamique libNETGEN.so va être construite à partir des librairies static de Netgen.

```
ifeq (@WITHNETGEN@,yes)
SBDIRS = OBJECT SMDS SMESHDS Driver DriverMED DriverDAT DriverUNV \
MEFISTO NETGEN SMESH SMESH_I SMESHFiltersSelection SMESHGUI \
SMESH_SWIG
endif
```

- Modification du fichier *SMESH_SRC/src/SMESH/Makefile.in* pour rajouter les fichiers qui wrappent les appels au mailleur tétraédrique de Netgen et les bonnes options de compilation/linkage.

```
ifeq (@WITHNETGEN@,yes)
EXPORT_HEADERS += SMESH_NETGEN_3D.hxx
LIB_SRC += SMESH_NETGEN_3D.cxx
NETGEN_INCLUDES=@NETGEN_INCLUDES@
CPPFLAGS += $(NETGEN_INCLUDES)
CXXFLAGS += $(NETGEN_INCLUDES)
LDFLAGS += -lNETGEN
endif
```

- Modification du fichier *SMESH_SRC/src/SMESH_I/Makefile.in*, équivalent CORBA de la précédente modification.

```

ifeq (@WITHNETGEN@,yes)
  LIB_SRC += SMESH_NETGEN_3D_i.cxx
  LDFLAGS += -lNETGEN
endif

```

- Modification du fichier *idl/SMESH_BasicHypothesis.idl* pour rajouter l'algorithme NETGEN_3D qui correspond au mailleur tétraédrique de Netgen.

```

interface SMESH_NETGEN_3D : SMESH_3D_Algo
{
};

```

- Modification du fichier *SMESH_SRC/src/SMESHGUI/SMESHGUI_icons.po* pour rajouter l'algorithme NETGEN_3D qui correspond au mailleur tétraédrique de Netgen dans la GUI de SMESH (bouton de sélection).

```

#mesh_tree_algo_tetra
msgid ICON_SMESH_TREE_ALGO_Tetra_3D
msgstr mesh_tree_algo_tetra.png

```

- Modification des fichiers *SMESH_SRC/ressources/SMESH_[en,fr].xml* pour rajouter les commentaires sur les boutons de la GUI de SMESH. Ces boutons permettent le choix du mailleur tétraédrique de Netgen et de la taille des éléments volumiques.

```

<popup-item item-id="5033" pos-id="" label-id="Max. Hexahedron or
Tetrahedron Volume" icon-id="mesh_hypo_volume.png" tooltip-id="" accel-
id="" toggle-id="" execute-action="" />
.
.
.
.
.
<popup-item item-id="5020" pos-id="" label-id="Hexahedron (i, j, k)"
icon-id="mesh_algo_hexa.png" tooltip-id="" accel-id="" toggle-id=""
execute-action="" />
<popup-item item-id="5021" pos-id="" label-id="Tetrahedron (Netgen)"
icon-id="mesh_algo_hexa.png" tooltip-id="" accel-id="" toggle-id=""
execute-action="" />
.
.
.
.
.
<toolbutton-item item-id="5033" label-id="Max. Hexahedron or
Tetrahedron Volume" icon-id="mesh_hypo_volume.png" tooltip-id=" Max.
Hexahedron or Tetrahedron Volume Hypothesis" accel-id="" toggle-id=""
execute-action="" />
.
.
.
.
.
<toolbutton-item item-id="5020" pos-id="" label-id="Hexahedron
(i, j, k)" icon-id="mesh_algo_hexa.png" tooltip-id=" Hexahedron (i, j, k)
Algorithm" accel-id="" toggle-id="" execute-action="" />
<toolbutton-item item-id="5021" pos-id="" label-id="Tetrahedron
(Netgen)" icon-id="mesh_algo_hexa.png" tooltip-id=" Tetrahedron (Netgen)
Algorithm" accel-id="" toggle-id="" execute-action="" />

```

Attention: un fichier .png *mesh_algo_tetra.png* doit être produit pour distinguer l'icône du mailleur hexahédrique de celui du mailleur tétraédrique (actuellement c'est une simple

copie).

- Modification du fichier *SMESH_SRC/src/SMESH/SMESH_HypothesisFactory.cxx* pour rajouter les nouvelles hypothèses concernant le choix du meilleur tétraédrique de Netgen et de la taille des éléments volumiques.

```
#include "SMESH_MaxElementVolume.hxx"
      .
      .
      .
#include "SMESH_NETGEN_3.hxx"
      .
      .
      .
creatorMap["MaxElementVolume"] =
      new SMESH_HypothesisCreator<SMESH_MaxElementVolume>;
      .
      .
      .
creatorMap["NETGEN_3D"] = new SMESH_HypothesisCreator<SMESH_NETGEN_3D>;
```

- Modification du fichier *SMESH_SRC/src/SMESH_I/SMESH_HypothesisFactory_i.cxx*, équivalent CORBA de la modification précédente.

```
#include "SMESH_MaxElementVolume_i.hxx"
      .
      .
      .
#include "SMESH_NETGEN_3_i.hxx"
      .
      .
      .
creatorMap["MaxElementVolume"] =
      new SMESH_HypothesisCreator_i<SMESH_MaxElementVolume_i>;
      .
      .
      .
creatorMap["NETGEN_3D"] =
      new SMESH_HypothesisCreator_i<SMESH_NETGEN_3D_i>;
```

- Modification du fichier *SMESH_SRC/src/SMESHGUI/SMESHGUI.cxx* pour rajouter les nouvelles hypothèses concernant le choix du meilleur tétraédrique de Netgen et de la taille des éléments volumiques dans la GUI de SMESH.

```
else if ( TypeAlgo.compare("NETGEN_3D" == 0)
      Hyp = myComponentMesh->CreateHypothesis(TypeAlgo, myStudyId );
      .
      .
      .
      tr("SMESH_MAX_ELEMENT_VOLUME_HYPOTHESIS"),
      .
      .
      .
      break;
    }
case 5021:
    {
      smeshGUI->CreateAlgorithm("NETGEN_3D", "Tetrahedral (Netgen)");
```

- Modification du fichier *SMESH_SRC/src/SMESH_I/SMESH_Gen_i.cxx*, équivalent CORBA pour le générateur.

```
#include "SMESH_MaxElementVolume_i.hxx"

        .
        .
        .
double length,maxElementsArea,maxElementsVolume;
        .
        .
        .
else if (strcmp(myHyp->GetName,"MaxElementVolume") == 0) {
    SMESH::SMESH_MaxElementVolume_var MEV =
        SMESH::SMESH_MaxElementVolume::_narrow( myHyp );
    maxElementVolume = MEV->GetMaxElementVolume();
    fprintf(destFile,"%f\n",maxElementVolume);
}
        .
        .
        .
double length,maxElementsArea,maxElementsVolume;
        .
        .
        .
else if (strcmp(aLine,"MaxElementVolume") == 0) {
    SMESH::SMESH_Hypothesis_var myHyp =
        this->CreateHypothesis(aLine,studyId);
    SMESH::SMESH_MaxElementVolume_var MEV =
        SMESH::SMESH_MaxElementVolume::_narrow( myHyp );
    fscanf(loadFile,"%s",aLine);
    maxElementVolume = atof(aLine);
    MEV->SetMaxElementVolume(maxElementVolume);
    string iorString = orb->objectto_string(MEV);
    sprintf(objectId,"%d",MEV->GetId());
    SMESHCorbaObj[string("Hypo_")+string(objectId)] = iorString;
}
}
```

- Modification des fichiers *SMESH_SRC/src/DriverMED/DriverMED_W_SMDS_Mesh.cxx* et *SMESH_SRC/src/DriverMED/DriverMED_W_SMESHDS_Mesh.cxx* pour l'écriture à travers le driver MED dans la structure de donnée SMDS et SMESHDS respectivement (ces structures sont internes au module SMESH).

```
case 4 :
    elem_Id[7].push_back(elem->GetID());
    nmailles[7]++;
    break;
}
```

- Modification du fichier *SMESH_SRC/src/DriverMED/DriverMED_R_SMDS_Mesh.cxx* pour la lecture à travers le driver MED sur la structure de donnée du maillage SMDS.

```

case MED_TETRA4 : {
  if (inuele) {
    for (j=0;j<nmailles[i];j++) {
      elem_id=(numele+j);
      ok = myMesh->AddVolumeWithID(*(connectivite+j*(taille)),
                                   *(connectivite+j*(taille)+1),
                                   *(connectivite+j*(taille)+2),
                                   *(connectivite+j*(taille)+3),
                                   elem_id);
    }
  }
  else {
    for (j=0;j<nmailles[i];j++) {
      cmpt;
      ok = myMesh->AddVolumeWithID(*(connectivite+j*(taille)),
                                   *(connectivite+j*(taille)+1),
                                   *(connectivite+j*(taille)+2),
                                   *(connectivite+j*(taille)+3),
                                   cmpt);
    }
    break;
  }
}

```

Modification du fichier *SMESH_SRC/src/DriverMED/DriverMED_R_SMESHDS_Mesh.cxx* pour la lecture à travers le driver MED sur la structure de donnée du maillage SMESHDS.

```

case MED_TETRA4 : {
  if (inuele) {
    for (j=0;j<nmailles[i];j++) {
      elem_id=(numele+j);
      ok = mySMESHDSMesh->AddVolumeWithID(
                                   *(connectivite+j*(taille-nsup)),
                                   *(connectivite+j*(taille-nsup)+1),
                                   *(connectivite+j*(taille-nsup)+2),
                                   *(connectivite+j*(taille-nsup)+3),
                                   elem_id);
    }
  }
  else {
    for (j=0;j<nmailles[i];j++) {
      cmpt;
      ok = mySMESHDSMesh->AddVolumeWithID(
                                   *(connectivite+j*(taille)),
                                   *(connectivite+j*(taille)+1),
                                   *(connectivite+j*(taille)+2),
                                   *(connectivite+j*(taille)+3),
                                   cmpt);
    }
    break;
  }
}

```

- Modification du fichier *SMESH_SRC/src/SMESH_SWIG/Makefile.in*, pour exporter les scripts python de teste du mailleur tétrahédrique: *SMESH_box_tetra.py*, *SMESH_box2_tetra.py*, *SMESH_box3_tetra.py*, *SMESH_mechanic_tetra.py*, *SMESH_fixation_tetra.py* et *SMESH_Partition1_tetra.py*. *SMESH_fixation_hexa.py* est un autre script python de teste du mailleur hexahédrique tandis que *SMESH_flight_skin.py* est un autre script python de teste du mailleur surfacique en triangle utilisant *MEFISTO_2D* avec l'hypothèse de maillage *SMESH_LengthFromEdges*. D'autres fichiers de teste sont prévus *SMESH_Partition[2,3,4,5].py* mais ceux-ci nécessitent un ajustement des paramètres de maillage.

```
EXPORT_PYSCRIPTS += SMESH_fixation_hexa.py, SMESH_flight_skin.py

ifeq (@WITHNETGEN@,yes)
  EXPORT_PYSCRIPTS += SMESH_box_tetra.py, SMESH_box2_tetra.py, \
                      SMESH_box3_tetra.py, SMESH_fixation_tetra.py, \
                      SMESH_Partition1_tetra.py
endif
```

- **Finalemment :**

```
./build_configure
cd SMESH_BUILD
rm config.*
SALOME2_ROOT/SMESH_SRC/configure -with-netgen=netgen_installation_path \
-prefix=smesh_install_path
make
```

où smesh_install_path est la directory d'installation du module SMESH.