How to use Octree Mesher via Makefiles for generating capsid meshes in order to perform FEM analysis

Jorge López, Salvador Botello, Rafael Herrera, and Mauricio Carrillo

Centro de Investigación en Matemáticas A.C.,
Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional
e-mail:jorge.lopez,botello,rherrera@cimat.mx,trippm@gmail.com

1 Introduction

This document explains the way to use the octree mesher via makefiles to perform numerical simulations of viral capsids by using the Finite Element Method (FEM).

The whole application is composed of five different sub-applications or programs, which can be used as a composite application via the main makefile, or sub-applications can be used as separate applications via the individual makefile. This document describes both approaches so that the user selects the way to use the application(s).

In the document, relating to the routes of the document, the letters `WD` is the working directory on the local machine that must be provided by the user.

2 Approach as individual applications

The code is composed of five applications which are:

- *CLEANER:* This application receives as input a ***.vdb*** file downloaded from the Virus Particle Explorer (*http://viperdb.scripps.edu/*) and removes all information not required by the mesher, all lines do not begin with word "ATOM".
- *MESHER:* This application generates the geometrical representation of the viral capsid and applies the boundary conditions with the given parameters, the input data is a spherical mesh given in format *vdb* and other parameters for imposing boundary conditions, such arguments will be explained in the following subsections.
- *ROTATOR:* This is the required transformation to move from the shear simulation to the nanoindentation simulation, this program merely requires the name of the file that contains the geometrical information of the capsid, input and output arguments are explained in following subsections.
- *SOLVER:* This application contains the solver for the equations system formed with the capsid data.

– *CREATES_PDB:* This application takes the obtained results with the solver and includes the displacements magnitude and vonmises in the *PDB* format by using occupancy and temperature factor columns, respectively.

The following subsections will explain the way to use each application as an individual application and its specific details like arguments, input and output file formats will be given.

2.1 CLEANER

This application takes the ***.vdb*** file downloaded from the Virus Particle Explorer and removes all not required information by the mesher, lines not beginning with "ATOM" are completely removed, and a new ***.vdb*** is created.

The makefile for this application is located on route `WD/CODE/CLEANER/makefile`, it can execute the following instructions:

– *make all:* Compiles the application.
– *make clean:* Removes all files created and only leaves the files required to compile and execute the mesher.
– *make run:* Executes a preset example (this line can be changed to clean another file).

The application requires two arguments to be executed, and such parameters are:

1. The name of the input ***.vdb*** file.
2. The name of the output ***.vdb*** file.

Both files (input and output) must accomplish the format given in the following link *https://www.cgl.ucsf.edu/chimera/docs/UsersGuide/tutorials/pdbintro.html*

2.2 MESHER

This is the mesher that converts the atoms of the capsid to a geometrical representation that can be used to perform numerical simulations via FEM by using hexahedrons.

The makefile for this application is located on route `WD/CODE/MESHER/makefile`, it can execute the following instructions:

– *make all:* Compiles the application.
– *make clean:* Removes all files created and only leaves the required files to compile and execute the mesher.
– *make run:* Executes a preset example (this line can be changed to generate a different mesh).

The mesher application requires 10 arguments to be executed, such parameters are:

1. The name of the *vdb* file, it contains the atoms represented as spheres and this argument is represented as a string.
2. The kind of virus, it is an int number indicating the kind of virus to mesh (1 for T1, 3 for T3, etc.).
3. This is an integer that can be one or two, 1 means Van Der Waals (VDW) solution and 2 means solvent-accessible surfaces (SAS). Actually, is only available option 1, future developments will include code to solve option 2.
4. Mesh resolution in Å, represented as a double value.
5. Fold of capsid to be aligned with the Y-axis, actually are only valid values 2,3 and 5 represented as an integer number.
6. Index of the fold to align with the Y-axis, it is an integer from 0 to 11.
7. Virus name in capital letter represented as a string.
8. Cone amplitude to set boundary conditions measured in degrees, the cone has the origin in the center of the capsid and drives through the fold aligned with Y-axis, it is a double value.
9. Variation in the proportion of loaded elements, it is a double value positive if it is required to set more loaded elements, negative is the opposite and zero means no change. This parameter is used to have control of the loaded elements.
10. Young modulus to be used in the simulation, this is a double value. NOTE: The value used must be divided by 10,000 because the units are modified to prevent mistakes while generating the mesh.

The input of this application is the ***.vdb*** file cleaned by using CLEANER application and it has many output files which are described right away:

- **.geometry.dat:** File containing the geometry information for performing the numerical simulation by using FEMT (*http://personal.cimat.mx:8181/ miguel-vargas/FEMT/*).
- **.problem.dat:** File containing the problem information for performing the numerical simulation by using FEMT.
- **.solver.dat:** File containing the solver information for performing the numerical simulation by using FEMT.
- **capside_mesh.msh:** File containing the information of capsid atoms in GiD (*https://www.gidhome.com/*) format, represented by a spherical mesh.
- **warnings.log:** File containing the list of ATOMS not used by the mesher because are not valid atoms of them belong to not valid aminoacids

2.3 ROTATOR

This is the application that can be used to rotate the mesh generated with the MESHER application to change the shear simulation by the nanoindentation simulation. This application is only used in case the user requires executing a nanoindentation simulation.

The makefile used to compile the code used to rotate the mesh is located on the route `WD/CODE/ROTATOR/makefile`, this makefile can execute the following instructions:

- *make all:* Compiles the application.
- *make clean:* Removes all files created and leaves only the files required to compile and execute the ROTATOR.
- *make run:* Executes a preset example (this line can be changed to rotate other mesh).

The mesher requires two arguments to be executed, such parameters are:

1. This is the name of the file that contains the geometry information in the format required by FEMT (`http://personal.cimat.mx:8181/~miguelvargas/FEMT/`), this is a string.
2. This is the name of the file to save the rotated geometry in the format required by FEMT (`http://personal.cimat.mx:8181/~miguelvargas/FEMT/`), this is a string and can maintain the same name than the previous argument.

2.4 SOLVER

This is the application used to create the system of equations and it also solves the system and saves the required files to visualize the results on GiD (`https://www.gidhome.com/`). The application is contained in the FEMT software (`http://personal.cimat.mx:8181/~miguelvargas/FEMT/`), it contains several applications to solve systems of equations and even software to execute numerical simulations.

The makefile for this application is located on route `WD/CODE/SOLVER/build/gcc/Makefile`, this makefile can execute the following instructions:

- *make all:* Compiles all applications.
- *make clean:* Removes all files created and leaves only the files required to compile and execute the SOLVER.

FEMT contains many applications and in this case is only used the Solid application its solver is located on the route `WD/CODE/SOLVER/gid/problemtypes/Solid.gid/Solid`, this Solver requires two arguments to be executed which are:

- The name of the files that contain the solver, problem and geometry data. NOTE: It is only required the name of the files without extension.
- The log level for the messages printed on the console.
  - 0: Means to print only fatal error messages.
  - 1: Means to print the description of the current process.
  - 2: Solver iterations are also printed.

The first argument of this application is a string with the name of the problem represented by three files with the same name but different extension, for instance if the problem to solve is named *simulation_example* then this is the name that must be used in the command line to execute the solver, the name must be in three files named as follows:

- *simulation_example*.**problem.dat**: This file contains the definition of the problem, as material properties, boundary conditions, and the variables to be solved in the problem.
- *simulation_example*.**solver.dat**: This file contains the solver parameters like preconditioner, number of parallel threads, tolerance of solution, number of iterations, etc.
- *simulation_example*.**geometry.dat**: This file contains the geometry information.

2.5 CREATES_PDB

This is the application that takes the results in the GiD format with extension *.post.msh* and *.post.res* and translate the results on nodes and gauss points to the center of the atoms in the *vdb* file so that the magnitude of the displacement is printed on columns Occupancy(55-60) and the vonmises is printed on columns Temperature factor(61-66)

The makefile for this application is located on route `WD/CODE/CREATES\_PDB/makefile`, this makefile can execute the following instructions:

- *make all:* Compiles the application.
- *make clean:* Removes all files created and leaves only the files required to compile and execute the application.
- *make run:* Executes a preset example (This line can be changed in order to generate other *pdb*).

The application requires five arguments to be executed, which are:

1. The name of the *vdb* file that contains the information of the atoms. This is a string
2. The name of the mesh file used to observe the results on the GiD software. This is a string.
3. The name of the results file in the GiD format. This is a string.
4. The name of the *pdb* file to save the results of displacement and vonmises on the atoms. This is a string.
5. An integer name with the refinement level to use in the octree for searching the element that contains the atom center to interpolate the results.

The following section utilizes the information of the applications explained in the previous lines and presents it as a whole.

3 Approach as a whole application

In this case, one makefile is functional to compile all applications and then can be used to execute two kinds of simulations. The first one is to simulate the deformation produced by a shear force applied on the capsid, the other one is

for performing a simulation to produce a nanoindentation on the capsid. Each simulation can be customized to use different viruses or geometrical parameters.

The makefile route to treat the code as a whole is:
`WD/CODE/makefile`
This makefile has a series of instructions that can be executed such as:

- *make all:* Compiles all individual applications.
- *make clean:* Removes all files generated and leaves the ones needed to compile.
- *make shear:* Execute the simulation to produce a deformation via applying shear force.
- *make nanoindentation:* Execute the simulation when the force is in the same direction that the fold aligned with the Y-axis and it produces a nanoindentation.

Both simulations presented are customizable in the sense that the arguments of the program can be changed in order to obtain other results. The way to customize each application is based on the information presented in the previous section where each application is explained as an individual application.