ppOpen-HPC:
Open Source Infrastructure for Development and Execution of Large-Scale Scientific Applications on Post-Peta-Scale Supercomputers with Automatic Tuning (AT).

ppOpen-APPL/DEM

ver. 1.0.0

User's guide

License

This software is one of the results of JST CREST project “ppOpen-HPC: Open Source Infrastructure for Development and Execution of Large-Scale Scientific Application on Post-Peta-Scale Supercomputers with Automatic Tuning (AT)” project.

Table of contents

1. Introduction 4

1.1 ppOpen-APPL/DEM and the discrete element method 4

1.2 ppOpen-APPL/DEM-Util 4

2. Installation and Quick Start 9

2.1 ppohDEM_1.0.0.tar 9

2.2 Structure of Directories 9

2.3 Quick Start 10

1.1.1. Preparation 10

1.1.2. Modify ‘Makefile.in’ 10

1.1.3. Compile/install ppOpen-APPL/DEM-Util 10

1.1.4. Compile/install ppOpen-APPL/DEM and a sample code 11

1.1.5. Set input files 11

1.1.6. Running the code 17

1.1.7. Clean/uninstall 17

3. ppOpen-APPL/DEM 18

3.1 Structure 18

3.2 Modules 20

3.2.1. ppOpen-APPL/DEM-Util 20

3.2.2. ppOpen-APPL/DEM 23

3.3 Subroutines 28

3.3.1 ppOpen-APPL/DEM-Util 28

3.3.2 ppOpen-APPL/DEM 43

1. Introduction

1.1 ppOpen-APPL/DEM and the discrete element method

ppOpen-APPL/DEM is a set of open libraries that provides a fundamental tool to perform a numerical simulation of an ensemble of large numbers of particles. A particle method is any of a family of numerical methods to compute the motion of a large number of particles. In a particle method, the evolution of system properties is simulated through the trajectories and evolution of the particles' properties representing the system's physical properties. Because simulating the motion of interacting particles is simple, particle methods are applied to various fields, such as molecular bioscience, material science, civil engineering, oceanography, and astrophysics. The Discrete Element Method (DEM) is one of the particle method. In DEM simulations, the particles are assumed to have short length dissipative interactions and not to have stateful contacts and complicated geometries, so that the DEM method computes the motion of large numbers of particles efficiently. Therefore, the DEM widely accepted as a practical numerical method for engineering purposes including powder mechanics, rock mechanics and earthquake mechanics as well as granular flows. The ppOpen-APPL/DEM libraries are based on the standard DEM algorithm and developed with the massively parallel computing techniques. We expect these libraries will help the users who develop the DEM application programs.

1.2 ppOpen-APPL/DEM-Util

ppOpen-APPL/DEM-Util is a set of utility libraries that provides useful tools for application programs of the DEM. The ppOpen-APPL/DEM-Util libraries mainly consist of two main modules, ppOpen-APPL/DEM-Util-distance_calculate and ppOpen-APPL/DEM-Util-objects_update. Figure 1.1 shows the constitutions of the The ppOpen-APPL/DEM libraries.

The ppOpen-APPL/DEM-Util-distance_calculate library is a pre-processing utility program to prepare the 3D objects data for the computation of the DEM application program from the stereolithography (STL) format data files. An STL file consists of a list of triangle meshes (facets), and each triangle mesh is described by three vertices. List 1.1 shows the example of the STL format data file.

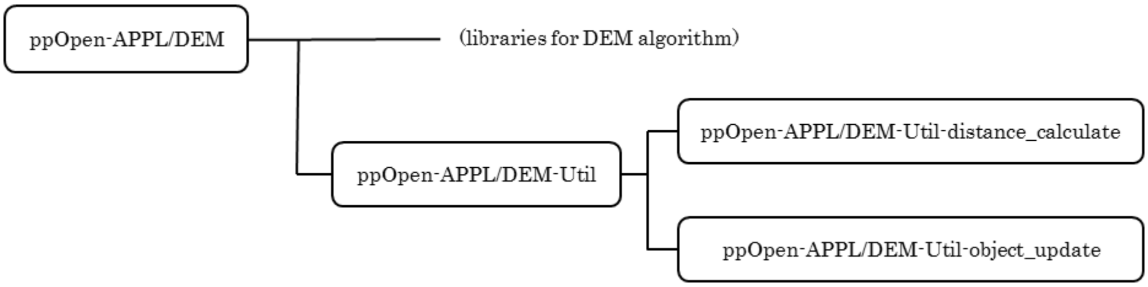
ppOpen-APPL/DEM
(libraries for DEM algorithm)
ppOpen-APPL/DEM-Util
ppOpen-APPL/DEM-Util-distance_calculate
ppOpen-APPL/DEM-Util-object_update

Figure 1.1: Constitutions of the The ppOpen-PPL/DEM ibraries.

```
solid STL generated by MeshLab
  facet normal -1.452403e-001 -2.377237e-001 -9.604128e-001
    outer loop
      vertex   3.303870e-002 5.661200e-002 -5.012260e-002
      vertex   3.210190e-002 5.752250e-002 -5.020630e-002
      vertex   3.305460e-002 5.750240e-002 -5.034540e-002
    endloop
  endfacet
  facet normal 2.252038e-001 1.431148e-001 -9.637435e-001
    outer loop
      vertex   3.303870e-002 5.661200e-002 -5.012260e-002
      vertex   3.219810e-002 5.657380e-002 -5.032470e-002
      vertex   3.210190e-002 5.752250e-002 -5.020630e-002
    endloop
  endfacet
  facet normal -2.924500e-001 6.079650e-002 -9.543463e-001
    outer loop
      vertex   3.065800e-002 5.622500e-002 -4.995850e-002
      vertex   3.129200e-002 5.657010e-002 -5.013080e-002
      vertex   3.061270e-002 5.582190e-002 -4.997030e-002
    endloop
  endfacet
…
endsolid vcg
```

List 1.1: Example of an STL format data file.

A subroutine in the ppOpen-APPL/DEM-Util-distance_calculate library reads an STL data file, and sets the three dimensional objects data in the computational space of the DEM application program. Figure 1.2 and Figure 1.3 show an example of three dimensional objects data for the DEM application program and a sample triangulated surface of a three dimensional object from an STL data file.

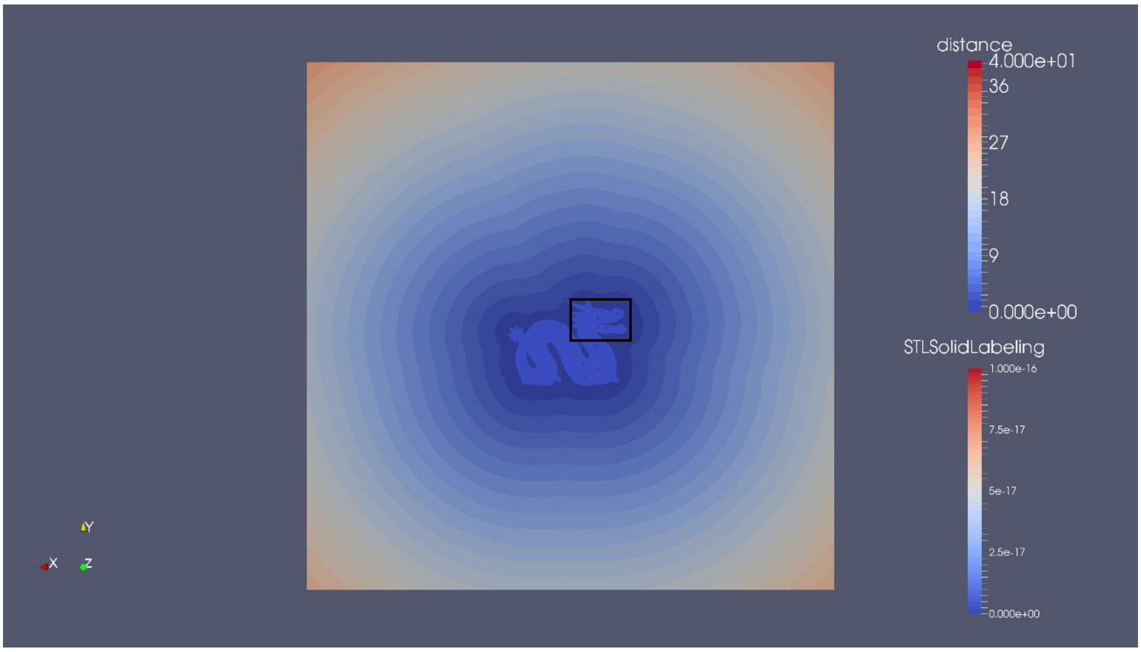

Figure 1.2: Example of three dimensional objects data for the DEM application program and a sample triangulated surface of a three dimensional (3D) object from an STL data file. The sample 3D object data is downloaded from the Stanford 3D scanning repository: http://graphics.stanford.edu/data/3Dscanrep/, and converted to an STL data file.

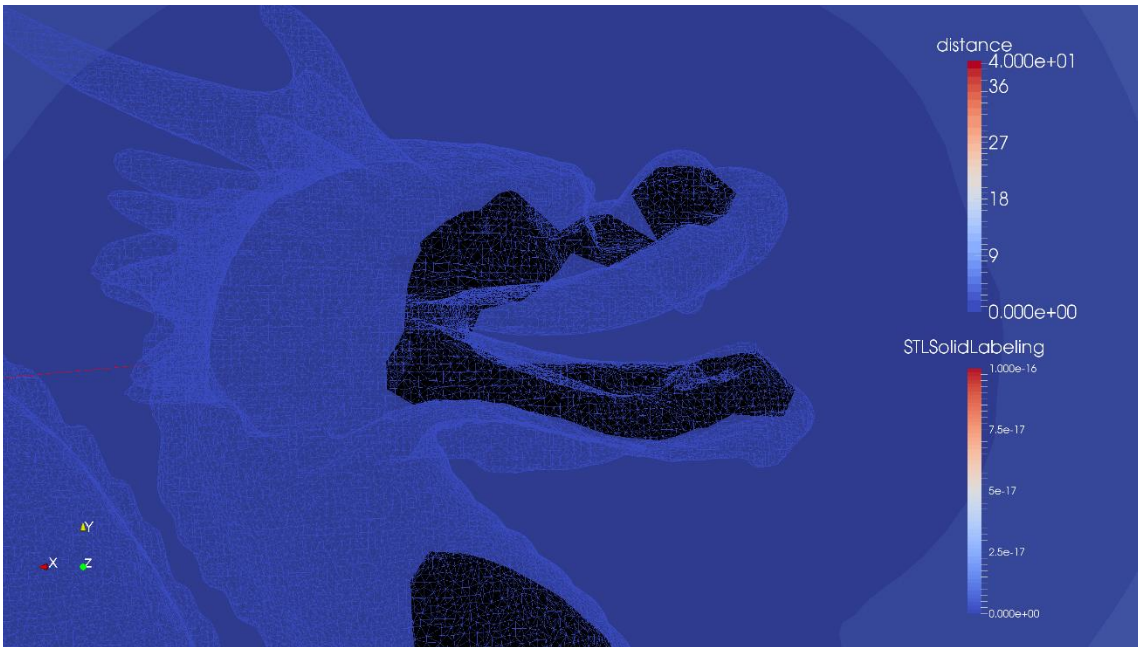

Figure 1.3: Enlargement image of three dimensional objects data and a triangulated surface of a three dimensional object.

In the library, the subroutine calculates the distance between each coordinate of the computational space outside of a three dimensional object and the surface of the object, and then the distance data is passed to the DEM application program, so that the DEM application program can utilise this distance data for the computation instead of using the STL triangulated surface data of a three dimensional object.

The ppOpen-APPL/DEM-Util-objects_update library is the other main module of the utility library that updates the position of the three dimensional objects in the computational space in the DEM application program. With this objects_update library, the objects move forward and backward, left and right, and up and down in the computational space of the DEM application program. In addition, the objects can rotate using quaternion rotations. A subroutine of the ppOpen-APPL/DEM-Util-objects_update library reads the distance data produced by the ppOpen-APPL/DEM-Util-distance_calculate library, and then the distance data is rearranged and interpolated by trilinear interpolation. After this distance data conversion, the distance data is passed to the main DEM library programs and utilised for the computation of the DEM. Figure 1.4 shows the procedures of the ppOpen-APPL/DEM-Util library programs. Both libraries can be easily adapted to application programs of the DEM, and these libraries will help the users who develop the DEM application programs.

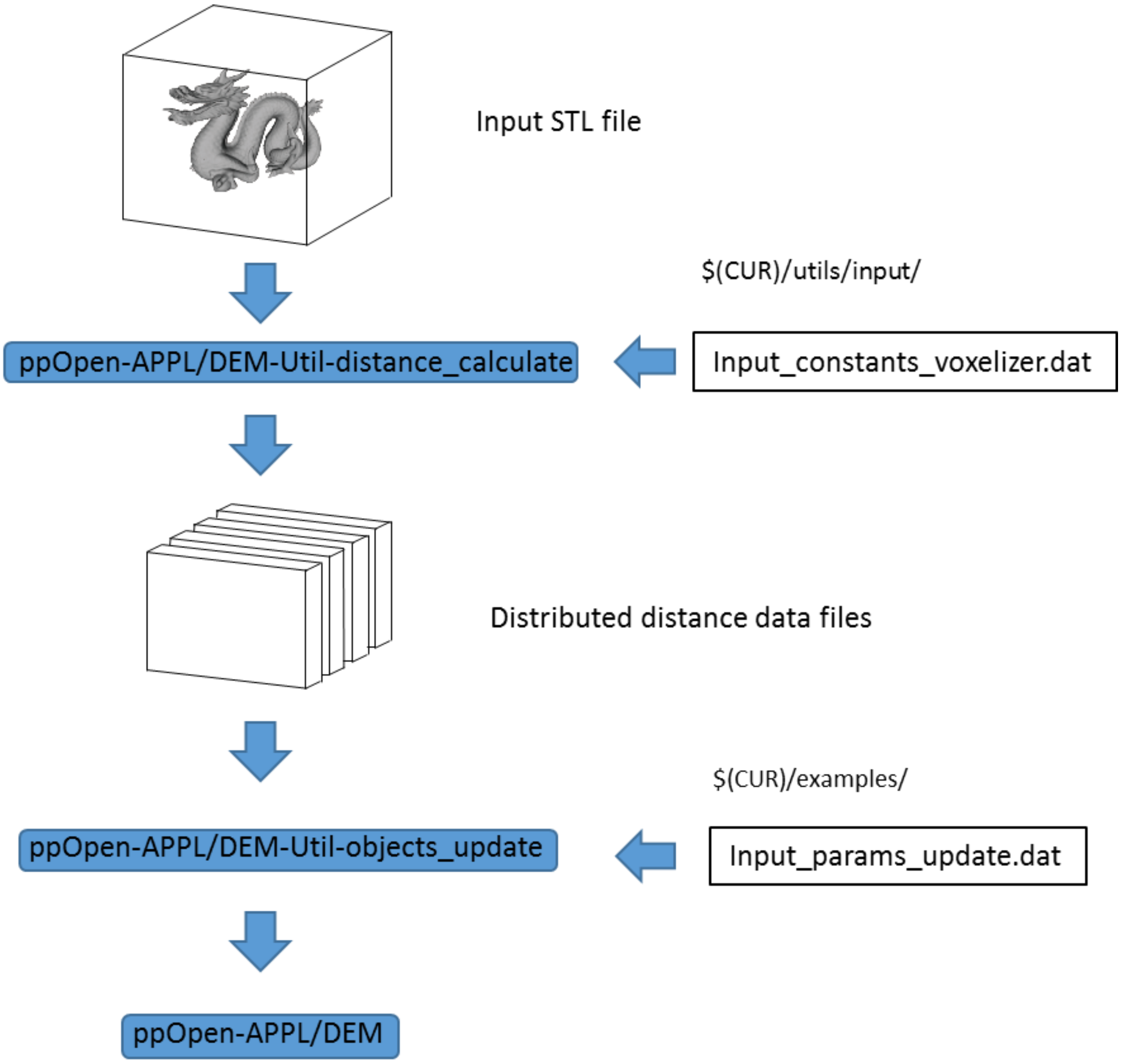
Input STL file
$(CUR)/utils/input/
ppOpen-APPL/DEM-Util-distance_calculate
Input_constants_voxelizer.dat
Distributed distance data files
$(CUR)/examples/
ppOpen-APPL/DEM-Util-objects_update
Input_params_update.dat
ppOpen-APPL/DEM

Figure 1.4: Procedures of the ppOpen-APPL/DEM-Util library programs.

2. Installation and Quick Start

2.1 ppohDEM_1.0.0.tar

The “ppohDEM_1.0.0.tar” archive file includes the followings:

- Source code files of “ppOpen-APPL/DEM ver. 1.0.0”
- Source code files of “ppOpen-APPL/DEM-Util”
- Source code files of sample programs
- Sample Makefiles
- Sample data files for “ppOpen-APPL/DEM-Util”

2.2 Structure of Directories

The “ppohDEM_1.0.0.tar” archive includes the following directories. $(CUR) denotes the directory where the “ppohDEM_1.0.0.tar” archive is unpacked.

Name of Directory	Contents
$(CUR)/src	source code files of ppOpen-APPL/DEM
$(CUR)/utils	source code files of ppOpen-APPL/DEM-Util
$(CUR)/utils/data	sample data sets of ppOpen-APPL/DEM-Util
$(CUR)/utils/input	sample control data of ppOpen-APPL/DEM-Util
$(CUR)/examples	sample source code files using ppOpen-APPL/DEM
$(CUR)/examples/data	Particles output files
$(CUR)/examples/obj	Objects output files
$(CUR)/examples/pov	POV-Ray output files
$(CUR)/include	directory that stores created module files
$(CUR)/lib	directory that stores created libraries
$(CUR)/bin	directory that stores created executable programs
$(CUR)/doc	documents
$(CUR)/etc	examples of ‘Makefile.in’

2.3 Quick Start

1.1.1. Preparation

- Fortran 90 compilers (Operations have been confirmed with GNU Fortran and Fujitsu compilers)
- MPI and OpenMP libraries

1.1.2. Modify 'Makefile.in'

Samples of 'Makefile.in' are found in $(CUR)/etc.

Samples of 'Makefile.in'	Compiler
$(CUR)/etc/Makefile.in.gfortran	GNU Fortran
$(CUR)/etc/Makefile.in.fx10	Fujitsu FX10

1.1.3. Compile/install ppOpen-APPL/DEM-Util

Operations	Files created (libraries, module files, exec. files)
$> cd $(CUR) $> make clean_util	
$> make util	$(CUR)/lib/ppohDEM_Util_Lib.a $(CUR)/bin/sample_DEMutil
$> make install_util	$(PREFIX)/lib/ppohDEM_Util_Lib.a $(PREFIX)/bin/ sample_DEMutil

1.1.4. Compile/install ppOpen-APPL/DEM and a sample code

Operations	Files created (libraries, module files, exec. files)
$> cd $(CUR) $> make clean	
$> make	$(CUR)/lib/ppohDEMlib.a $(CUR)/bin/sample_main01
$> make install	$(PREFIX)/lib/ppohDEMlib.a $(PREFIX)/bin/sample_main01

1.1.5. Set input files

The directories, $(CUR)/utils/input and $(CUR)/examples, include input files that specify parameters for running programs. input_constants_voxelizer.dat included in $(CUR)/utils/input is an input file that rules the action of ppOpen-APPL/DEM-Util-distance_calculate libraries. The ppOpen-APPL/DEM-Util-distance_calculate libraries produce distance data files, and output the data files to a directory specified by the input file.
The directory $(CUR)/examples includes input_params_update.dat that is the other input file that sets the parameters for the ppOpen-APPL/DEM sample program and the ppOpen-APPL/DEM-Util-objects_update libraries.

(A) input_constants_voxelizer.dat

input_constants_voxelizer.dat is the input file for the ppOpen-APPL/DEM-Util-distance_calculate libraries that specifies the input STL file name of the three dimensional (3D) objects, the name of the output directory, the number of the voxels in the x, y and z axis in the distance data space and the parameters used for the libraries, such as the length of the data space in the x, y and z directions, the size of the 3D objects and the centre position of the 3D objects in the data space. Figure 2.1 illustrates the dimensions of the parameters given by the input file.

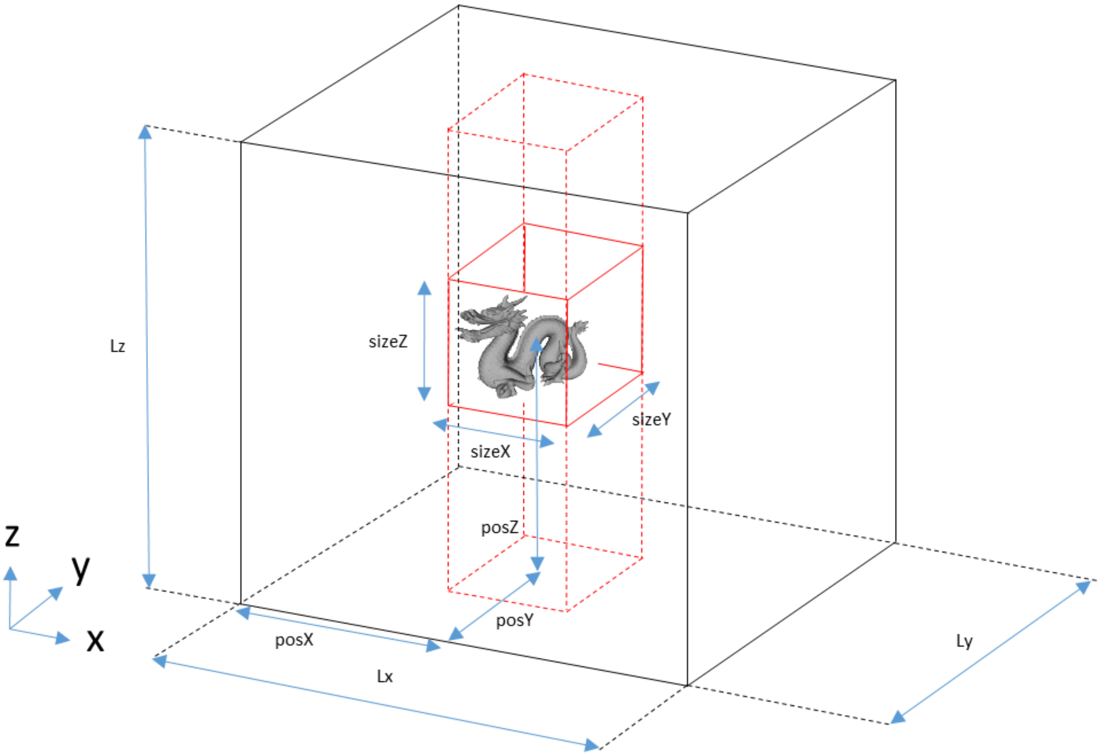

Figure 2.1: Dimensions of the parameters. Lx, Ly and Lz denote the length of the distance data space for each direction, sizeX, sizeY and sizeZ denote the size of the three dimensional (3D) objects, and posX, posY and posZ denote the centre position of the 3D objects.

The data format of input_constants_voxelizer.dat is as follows:

Line number	Description	Data type	Number of values
1	Input file name	Character	1
2	Output directory	Character	1
3	Number of voxels	Integer(kind=kint)	3
4	Length of the data space	Real(kind=kreal)	3
5	Size of the objects	Real(kind=kreal)	3
6	Position of the objects	Real(kind=kreal)	3

List 2.1 shows the sample file of input_constants_voxelizer.dat.

```
./data/dragon.stl
./
76 76 76
Lx_Ly_Lz: 44 44 44
sizeX_sizeY_sizeZ: 10 10 10
posX_posY_posZ: 22 22 11
```

List 2.1: Sample file of input_constants_voxelizer.dat.

(B) input_params_update.dat

input_params_update.dat is the input file for the ppOpen-APPL/DEM sample program and the ppOpen-APPL/DEM-Util-objects_update libraries that specifies the length of the computational space for each direction, the maximum time step and the parameters for the ppOpen-APPL/DEM-Util-objects_update libraries, such as the number of the voxels in the computational space, the start cell position of the computational space, the velocity of the three dimensional (3D) objects for each direction, the elements of quaternion rotation and the centroid of the 3D objects in the data space. Figure 2.2 shows the dimensions of the parameters given by the input file.

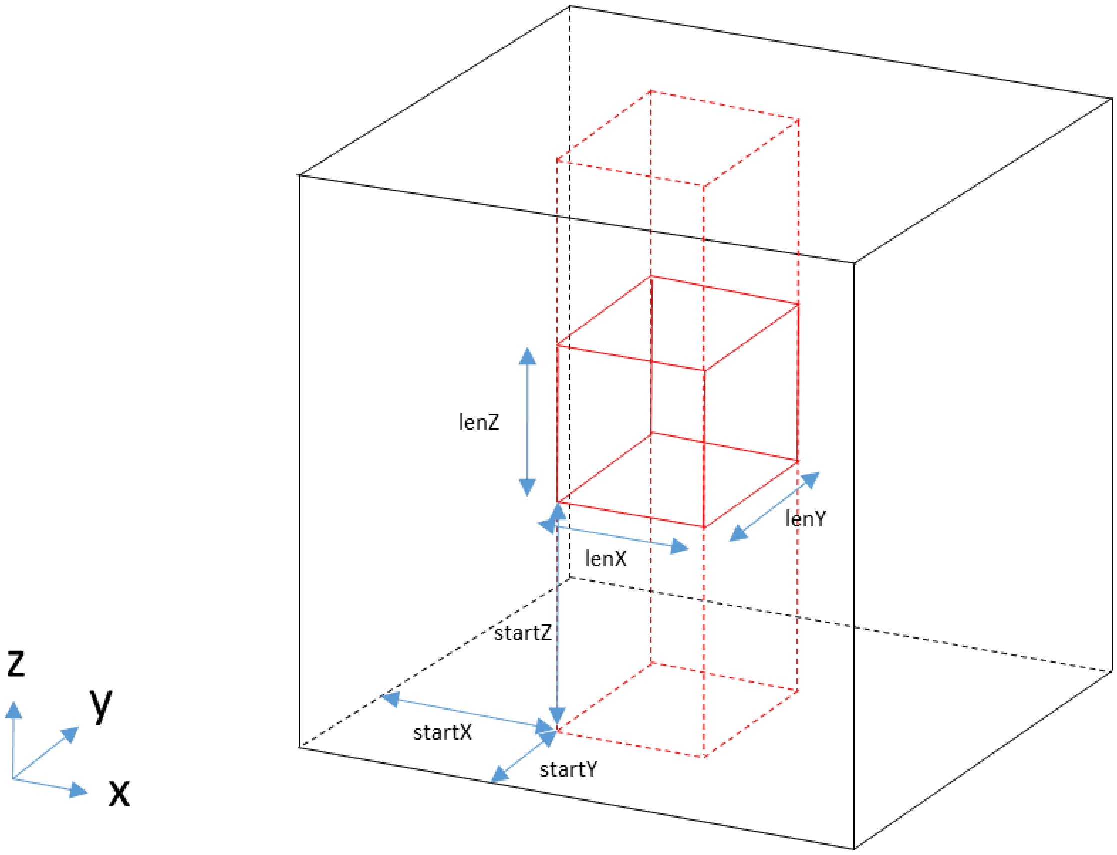

Figure 2.2: Dimensions of the parameters. lenX, lenY and lenZ denote the number of the voxels in the computational space, and startX, starty and startZ denote the start cell position of the computational space.

Note that the length of the computational space and the number of the voxels determine the resolution of the cells for the DEM computation. As this resolution of the cells in the computational space must be identical with that in the distance data space from the distance data file, the length of the computational space and the number of the voxels must be selected carefully.

The start cell position of the computational space defines the initial place of the computational space in the distance data space. The initial place of the computational space is used for the motion calculations of the computational space at each time step that determine the distance between the particles in the DEM computational space and the 3D objects. The motion calculation performs in two parts, namely, the translation and the quaternion rotation. Figure 2.3 and Figure 2.4 show the parameters for the translation and the quaternion rotation, respectively.

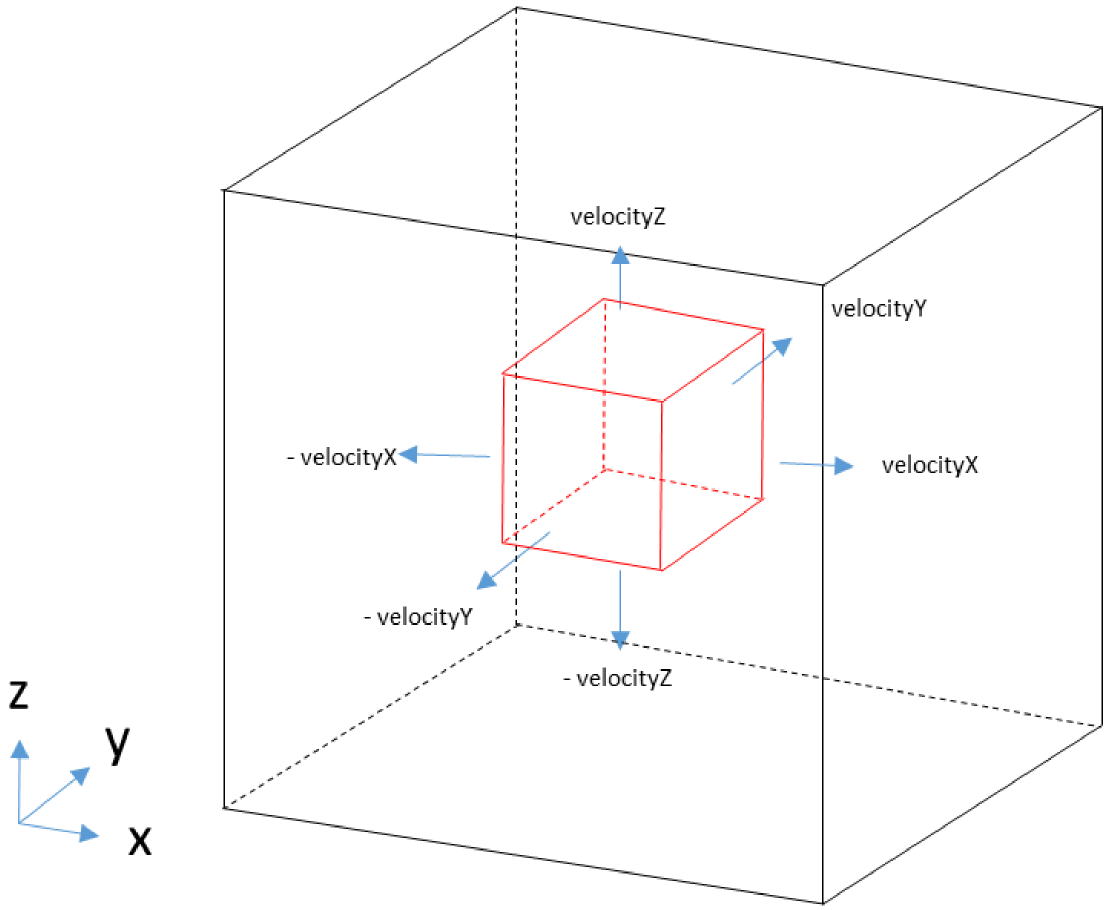

Figure 2.3: Parameters for the translation. velocityX, velocityY and velocityZ denote the velocity of the three dimensional objects for each direction.

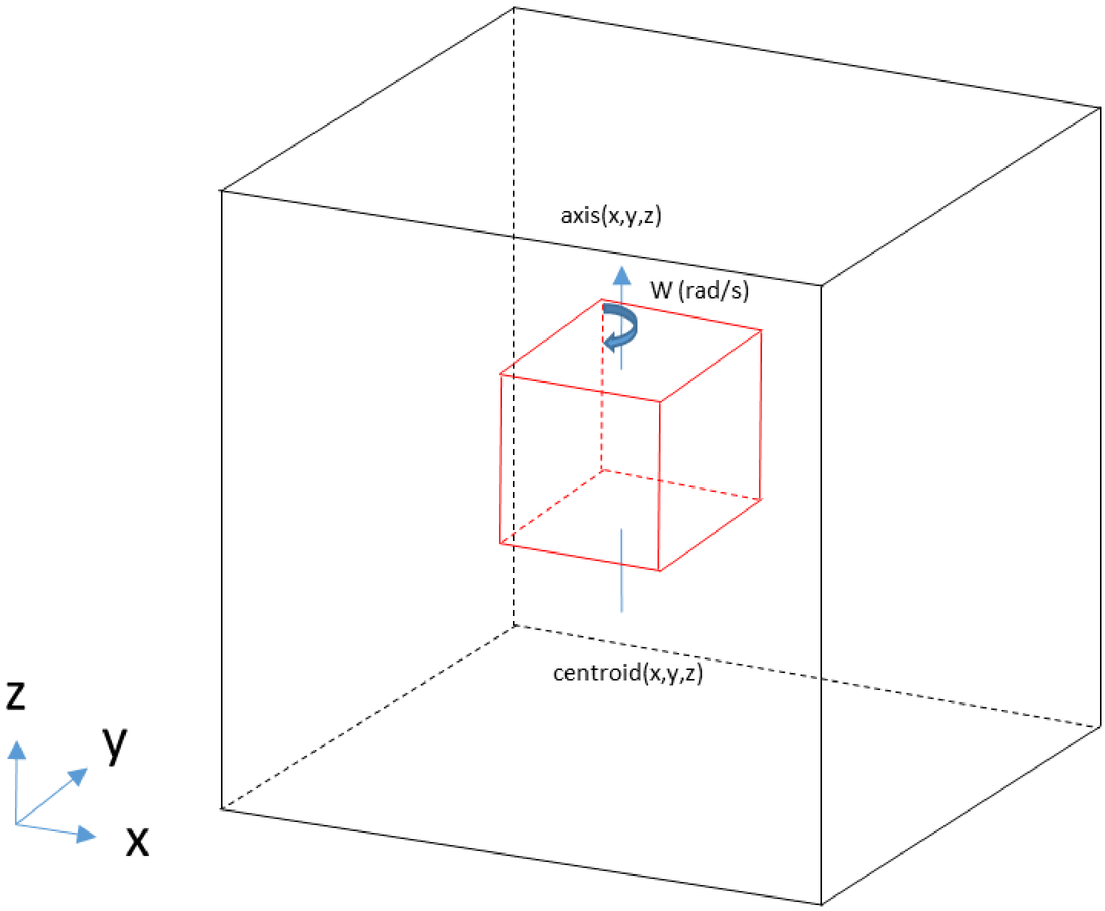

Figure 2.4: Parameters for the quaternion rotation. axis(x,y,z) denote the axis of the quaternion rotation, w denote the angler velocity of the quaternion rotation and centroid(x,y,z) denote the centroid of the quaternion rotation.

The data format of input_params_update.dat is as follows:

Line number	Description	Data type	Number of values
1	Length of the computational space	Real(kind=kreal)	3
2	Maximum time step	Real(kind=kreal)	1
3	Number of the voxels in the computational space	Integer(kind=kint)	3
4	The start cell position of the computational space	Integer(kind=kint)	3
5	Velocity of computational space for each direction	Real(kind=kreal)	3
6	The axis of the quaternion rotation	Real(kind=kreal)	3
7	The angler velocity of the quaternion rotation	Real(kind=kreal)	1
8	The centroid of quaternion rotation	Real(kind=kreal)	3

List 2.2 shows the sample file of input_params_update.dat.

```
Lx_Ly_Lz: 11 11 11
Time_finish: 300.
Obj_n_x_y_z: 20 20 20
Obj_winstart_x_y_z: 28 28 28
Obj_velocity_x_y_z: 0 0 0.005
Obj_omega_x_y_z: 0 0 1
Obj_omega_Pi_rad_per_w: 1200
Obj_centroid_x_y_z: 22 22 11
```

List 2.2: Sample file of input_params_update.dat.

1.1.6. Running the code

sample_DEMutil

```
$> cd $(CUR)/util
$> mpirun –np 4 ./sample_DEMutil
         With appropriate thread number for OpenMP (or corresponding operations).
```

sample_mail01

```
$> cd $(CUR)/examples
$> cp ../utils/distance_* ./            copy distance data files
$> cp ../utils/object_mesh.stl ./       copy the 3D objects file
$> ./sample_main01
```

1.1.7. Clean/uninstall

```
$> cd $(CUR)/
$> make clean           clean files.
$> make uninstall       delete all installed files and directories.
```

3. ppOpen-APPL/DEM

3.1 Structure

Figure 3.1 and Figure 3.2 show the structure of the ppOpen-APPL/DEM-Util library and that of the ppOpen-APPL/DEM library, respectively.

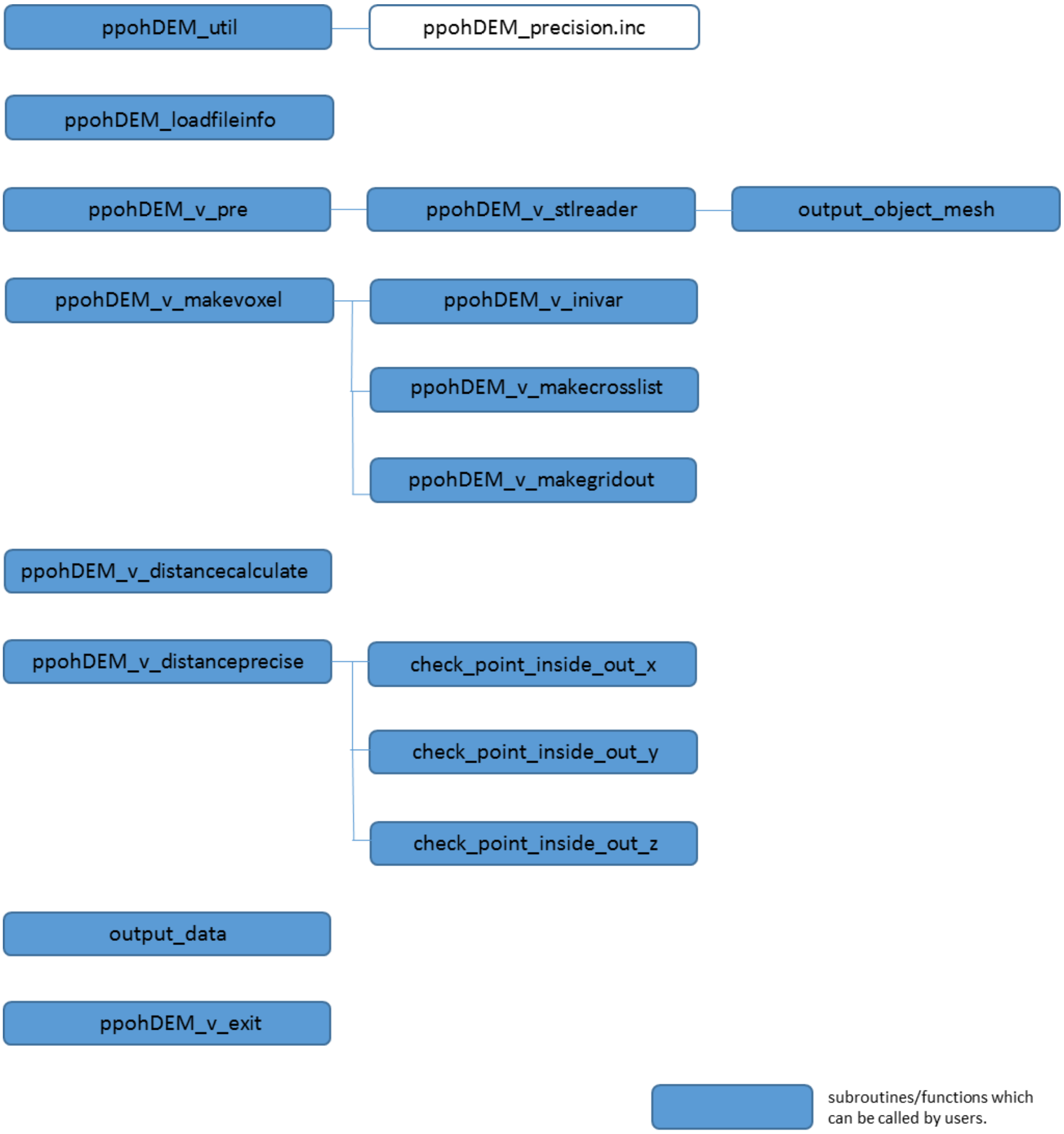

Figure 3.1: Structure of the ppOpen-APPL/DEM-Util library.

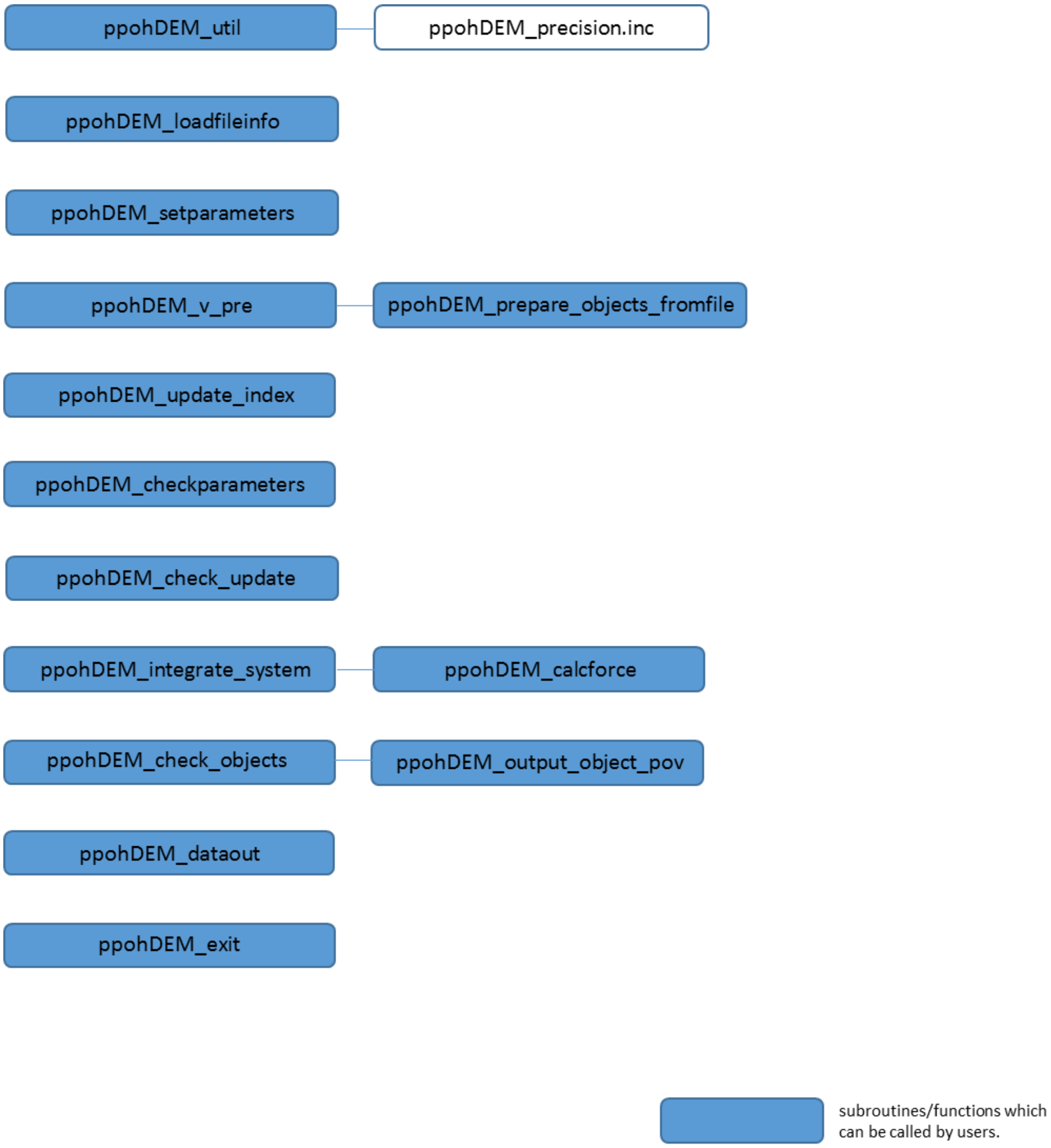

Figure 3.2: Structure of the ppOpen-APPL/DEM library.

3.2 Modules

3.2.1. ppOpen-APPL/DEM-Util

This module contains information on variables for meshes and voxels. The module is used by all subroutines in ppOpen-APPL/DEM-Util.

ppohDEM_precision.inc

```
        integer, parameter   ::  kint   = 4
        integer, parameter   ::  kreal = 8
        integer, parameter   ::  ppohDEM_name_length = 80
        real(4), parameter   :: FLT_MAX=+3.402823E+38
        real(4), parameter   :: FLT_MIN=-3.402823E+38
        real(8), parameter   :: DBL_MAX=+1.79769313486231D+308
        real(8), parameter   :: DBL_MIN=-1.79769313486231D+308
```

ppohDEM_util.f90

```
module ppohDEM_util

    implicit none
    public
    include 'mpif.h'
    include 'ppohDEM_precision.inc'

    type ppohDEM_fileinfo

        character(len=ppohDEM_name_length):: input_filename
        character(len=ppohDEM_name_length):: output_filename
    end type ppohDEM_fileinfo


    type ppohDEM_single_mesh

        type(ppohDEM_vec3) :: v1,v2,v3

    end type


    type ppohDEM_mesh

        type(ppohDEM_single_mesh),pointer :: mesh(:)

    end type


    type ppohDEM_voxel

        real(kind=kreal),pointer :: distance_3D(:,:,:)

        real(kind=kreal) :: Lx, Ly, Lz
        integer(kind=kint) :: cellX, cellY, cellZ
        real(kind=kreal) :: sizeX, sizeY, sizeZ
        real(kind=kreal) :: posX, posY, posZ

    end type


    type ppohDEM_gridco
        real(kind=kreal),pointer :: x(:),y(:),z(:)
    end type


end module ppohDEM_util
```

Parameters

kint = 4
kreal = 8
ppohDEM_name_length = 80
FLT_MAX=+3.402823E+38
FLT_MIN=-3.402823E+38
DBL_MAX=+1.79769313486231D+308
DBL_MIN=-1.79769313486231D+308

ppohDEM_fileinfo

input_filename: Input file name
output_filename: output file name

ppohDEM_single_mesh

v1: First vertex
v2: Second vertex
v3: Third vertex

ppohDEM_mesh

mesh: meshes

ppohDEM_voxel

distance_3D: Distance from the objects
L: Length of the distance data space
cell: Number of cells in the distance data space for each direction
size: Size of the objects

pos: Position of the objects in the distance data space

type ppohDEM_gridco

x: Number of cells for x direction
y: Number of cells for y direction
z: Number of cells for z direction

3.2.2. ppOpen-APPL/DEM

This module contains information on variables for particles, cells and objects. The module is used by all subroutines in ppOpen-APPL/DEM.

ppohDEM_precision.inc

```
  integer, parameter    ::  kint   = 4
  integer, parameter    ::  kreal = 8
  integer, parameter    ::  ppohDEM_name_length = 80
  real(4), parameter    :: FLT_MAX=+3.402823E+38
  real(4), parameter    :: FLT_MIN=-3.402823E+38
  real(8), parameter    :: DBL_MAX=+1.79769313486231D+308
  real(8), parameter    :: DBL_MIN=-1.79769313486231D+308
```

ppohDEM_util.f90

```
module ppohDEM_util

    implicit none
    public
```

```
include 'ppohDEM_precision.inc'


type ppohDEM_fileinfo
    character(len=ppohDEM_name_length):: input_filename
    character(len=ppohDEM_name_length):: output_filename
    character(len=ppohDEM_name_length):: input_params_filename

end type ppohDEM_fileinfo

type ppohDEM_particles
    integer(kind=kint) :: n
    real(kind=kreal), pointer :: radius(:)
    type(ppohDEM_rvec3), pointer :: pos(:)
    type(ppohDEM_rvec3), pointer :: vel(:)
    type(ppohDEM_rvec3), pointer :: force(:)
    type(ppohDEM_rvec3), pointer :: omega(:)
    type(ppohDEM_rvec3), pointer :: rforce(:)
    integer(kind=kint), pointer :: cellindex(:)
    integer(kind=kint), pointer :: newtoorig(:)
    type(ppohDEM_clist), pointer :: contact_list(:)
    type(ppohDEM_clist), pointer :: contact_list_wall(:)
end type

type ppohDEM_cells
    integer(kind=kint) :: n
    type(ppohDEM_rvec3) :: cellsize
    type(ppohDEM_ivec3) :: cellnum
    integer(kind=kint), pointer :: nei_cellindex(:)
    integer(kind=kint), pointer :: start_partindex(:)
    integer(kind=kint), pointer :: end_partindex(:)
end type

type ppohDEM_walls
    integer(kind=kint) :: n
    type(ppohDEM_rvec3), pointer :: point(:)
```

```
        type(ppohDEM_rvec3), pointer :: nvec(:)
    end type

    type ppohDEM_objects
        type(ppohDEM_ivec3) :: gridnum
        type(ppohDEM_rvec3) :: gridsize
        real(kind=kreal), pointer :: distance(:)
        type(ppohDEM_rvec3) :: vel
        type(ppohDEM_rvec4) :: omega
        type(ppohDEM_rvec3) :: centroid
        integer(kind=kint) :: n_mesh

    end type

    type ppohDEM_counter
        integer(kind=kint) :: itime
        real(kind=kreal) :: rtime
        integer(kind=kint) :: file_index
        real(kind=kreal) :: pos_accum
    end type

end module ppohDEM_util
```

Parameters

```
kint   = 4
kreal = 8
ppohDEM_name_length = 80
FLT_MAX=+3.402823E+38
FLT_MIN=-3.402823E+38
DBL_MAX=+1.79769313486231D+308
DBL_MIN=-1.79769313486231D+308
```

ppohDEM_fileinfo

input_filename: Input file name
output_filename: Output file name
input_params_filename: File name of input_params

ppohDEM_particles

n: Total number of particles
radius: Radius of particles
pos: Position of particles
vel: Velocity of particles
force: Force of particles
omega: Angular velocity of particles
rforce: Torque of particles
cellindex: Cell index
newtoorig: Original indices of particles
contact_list: Contact list
contact_list_wall: Contact list of walls

ppohDEM_cells

n: Total number of cells
cellsize: Size of cells
cellnum: Number of cells for each direction
nei_cellindex: Index of neighbour cells
start_partindex: Minimal particle index inside cell
end_partindex: Maximal particle index inside cell

ppohDEM_walls

n: Number of wall
point: Position of point that wall pass through

nvec: Normal vector of wall

ppohDEM_objects

gridnum: Number of grids for each direction
gridsize: Size of grids
distance: Distance from objects
vel: Velocity of motion
omega: Parameters for quaternion rotation
centroid: Centroid of quaternion rotation
n_mesh: Total number of mesh

ppohDEM_counter

itime: Integer elapsed time
rtime: Real elapsed time
file_index: File index
pos_accum: Distance that particle moves over

3.3 Subroutines

3.3.1 ppOpen-APPL/DEM-Util

ppohDEM_loadfileinfo

This subroutine reads the input file containing information for ppOpen-APPL/DEM-Util libraries.

```
subroutine ppohDEM_loadfileinfo (file_info,ftype)

    use ppohDEM_util

    type(ppohDEM_fileinfo) :: file_info
    integer(kind=kint) :: ftype
```

Parameters

file_info Derived Type: ppohDEM_fileinfo
ftype Derived Type: integer(kind=kint)

Uses the following Modules

ppohDEM_util

Called by the following subroutines in ppOpen-APPL/DEM-Util

(none)

Calls the following subroutines

(none)

ppohDEM_v_pre

This subroutine calls a subroutine for reading STL data files.

```
subroutine ppohDEM_v_pre (file_info,meshXYZ,voxel,gridco)

    use ppohDEM_util

    type(ppohDEM_fileinfo) :: file_info
    type(ppohDEM_mesh) :: meshXYZ
    type(ppohDEM_voxel) :: voxel
    type(ppohDEM_gridco) :: gridco
```

Parameters

file_info Derived Type: ppohDEM_fileinfo
meshXYZ Derived Type: ppohDEM_mesh
voxel Derived Type: ppohDEM_voxel
gridco Derived Type: ppohDEM_gridco

Uses the following Modules

ppohDEM_util

Called by the following subroutines in ppOpen-APPL/DEM-Util

(none)

Calles the following subroutines

ppohDEM_v_stlreader

ppohDEM_v_stlreader

This subroutine reads STL data files containing triangulated surfaces of three dimensional objects.

```
subroutine ppohDEM_v_stlreader (file_info,meshXYZ,voxel,gridco)

    use ppohDEM_util

    type(ppohDEM_fileinfo) :: file_info
    type(ppohDEM_mesh) :: meshXYZ
    type(ppohDEM_voxel) :: voxel
    type(ppohDEM_gridco) :: gridco
```

Parameters

file_info Derived Type: ppohDEM_fileinfo
meshXYZ Derived Type: ppohDEM_mesh
voxel Derived Type: ppohDEM_voxel
gridco Derived Type: ppohDEM_gridco

Uses the following Modules

ppohDEM_util

Called by the following subroutines in ppOpen-APPL/DEM-Util

ppohDEM_v_pre

Calles the following subroutines

output_object_mesh

output_object_mesh

This subroutine outputs modified STL data files used for ppOpen-APPL/DEM-Util-objects_update libraries.

```
subroutine output_object_mesh(file_info,meshXYZ)

  use ppohDEM_util

  type(ppohDEM_fileinfo) :: file_info
  type(ppohDEM_mesh) :: meshXYZ
```

Parameters

file_info Derived Type: ppohDEM_fileinfo
meshXYZ Derived Type: ppohDEM_mesh

Uses the following Modules

ppohDEM_util

Called by the following subroutines in ppOpen-APPL/DEM-Util

ppohDEM_v_stlreader

Calles the following subroutines

(none)

ppohDEM_v_makevoxel

This subroutine calls three subroutines for setting three dimensional objects in the distance data space, creating a cross list between voxels and three dimensional meshes and checking the voxels whether inside the three dimensional objects or not.

```
subroutine ppohDEM_v_makevoxel (file_info,meshXYZ,voxel,gridco)

    use ppohDEM_util

    type(ppohDEM_mesh) :: meshXYZ
    type(ppohDEM_voxel) :: voxel
    type(ppohDEM_gridco) :: gridco
```

Parameters

meshXYZ Derived Type: ppohDEM_mesh
voxel Derived Type: ppohDEM_voxel
gridco Derived Type: ppohDEM_gridco

Uses the following Modules

ppohDEM_util

Called by the following subroutines in ppOpen-APPL/DEM-Util

(none)

Calles the following subroutines

ppohDEM_v_inivar, ppohDEM_v_makecrosslist, ppohDEM_v_makegridout

ppohDEM_v_inivar

This subroutine sets three dimensional objects in the distance data space.

```
subroutine ppohDEM_v_inivar (file_info,meshXYZ,voxel,gridco)

    use ppohDEM_util

    type(ppohDEM_fileinfo) :: file_info
    type(ppohDEM_mesh) :: meshXYZ
    type(ppohDEM_voxel) :: voxel
    type(ppohDEM_gridco) :: gridco
```

Parameters

file_info Derived Type: ppohDEM_fileinfo
meshXYZ Derived Type: ppohDEM_mesh
voxel Derived Type: ppohDEM_voxel
gridco Derived Type: ppohDEM_gridco

Uses the following Modules

ppohDEM_util

Called by the following subroutines in ppOpen-APPL/DEM-Util

ppohDEM_v_makevoxel

Calles the following subroutines

(none)

ppohDEM_v_makecrosslist

This subroutine creates a cross list between voxels and three dimensional meshes.

```
Subroutine ppohDEM_v_makecrosslist (loopX,loopY,facet_crosslist,n_fcl,meshXYZ,gridco)

    use ppohDEM_util

    integer(kind=kint) loopX,loopY
    integer(kind=kint) :: facet_crosslist(*)
    integer(kind=kint) n_fcl
    type(ppohDEM_mesh) :: meshXYZ
    type(ppohDEM_gridco) :: gridco
```

Parameters

loopX,loopY Derived Type: integer(kind=kint)
facet_crosslist: Facet cross list
n_fcl: Number of facet cross list
meshXYZ Derived Type: ppohDEM_mesh
gridco Derived Type: ppohDEM_gridco

Uses the following Modules

ppohDEM_util

Called by the following subroutines in ppOpen-APPL/DEM-Util

ppohDEM_v_makevoxel

Calles the following subroutines

(none)

ppohDEM_v_makegridout

This subroutine determines whether each voxel is inside the three dimensional objects or not.

```
Subroutine ppohDEM_v_makegridout (loopX,loopY,facet_crosslist,n_fcl,meshXYZ,voxel,gridco)

    use ppohDEM_util

    integer(kind=kint) loopX,loopY
    integer(kind=kint) :: facet_crosslist(*)
    integer(kind=kint) n_fcl
    type(ppohDEM_mesh) :: meshXYZ
    type(ppohDEM_gridco) :: gridco
```

Parameters

loopX,loopY Derived Type: integer(kind=kint)
facet_crosslist: Facet cross list
n_fcl: Number of facet cross list
meshXYZ Derived Type: ppohDEM_mesh
gridco Derived Type: ppohDEM_gridco

Uses the following Modules

ppohDEM_util

Called by the following subroutines in ppOpen-APPL/DEM-Util

ppohDEM_v_makevoxel

Calles the following subroutines

(none)

ppohDEM_v_distancecalculate

This subroutine calculates distance between surfaces of the three dimensional objects and the voxels with Gauss-Seidel method.

```
subroutine ppohDEM_v_distancecalculate (voxel)

    use ppohDEM_util

    type(ppohDEM_voxel) :: voxel
```

Parameters

voxel Derived Type: ppohDEM_voxel

Uses the following Modules

ppohDEM_util

Called by the following subroutines in ppOpen-APPL/DEM-Util

(none)

Calles the following subroutines

(none)

ppohDEM_v_distanceprecise

This subroutine calculates accurate distance from the surfaces of the three dimensional objects to the close voxels.

```
subroutine ppohDEM_v_distanceprecise (voxel,meshXYZ,gridco)

    use ppohDEM_util

    type(ppohDEM_voxel) :: voxel
    type(ppohDEM_mesh) :: meshXYZ
    type(ppohDEM_gridco) :: gridco
```

Parameters

voxel Derived Type: ppohDEM_voxel
meshXYZ Derived Type: ppohDEM_mesh
gridco Derived Type: ppohDEM_gridco

Uses the following Modules

ppohDEM_util

Called by the following subroutines in ppOpen-APPL/DEM-Util

(none)

Calles the following subroutines

check_point_inside_out_x, check_point_inside_out_y, check_point_inside_out_z

check_point_inside_out_x

This subroutine determines whether each grid inside the three dimensional objects or not for the x direction.

```
subroutine check_point_inside_out_x (meshXYZ,voxel,gridco,grid,number_mesh,check_inside_x)

    use ppohDEM_util

    type(ppohDEM_mesh) :: meshXYZ
    type(ppohDEM_voxel) :: voxel
    type(ppohDEM_gridco) :: gridco
    type(ppohDEM_vec3) :: grid
    integer(kind=kint) number_mesh
    integer(kind=kint) :: check_inside_x
```

Parameters

meshXYZ Derived Type: ppohDEM_mesh
voxel Derived Type: ppohDEM_voxel
gridco Derived Type: ppohDEM_gridco
grid Derived Type: ppohDEM_vec3
number_mesh: Number of mesh
check_inside_x: Check inside or not for x direction

Uses the following Modules

ppohDEM_util

Called by the following subroutines in ppOpen-APPL/DEM-Util

ppohDEM_v_distanceprecise

Calles the following subroutines

(none)

check_point_inside_out_y

This subroutine determines whether each grid inside the three dimensional objects or not for the y direction.

```
subroutine check_point_inside_out_y (meshXYZ,voxel,gridco,grid,number_mesh,check_inside_y)

    use ppohDEM_util

    type(ppohDEM_mesh) :: meshXYZ
    type(ppohDEM_voxel) :: voxel
    type(ppohDEM_gridco) :: gridco
    type(ppohDEM_vec3) :: grid
    integer(kind=kint) number_mesh
    integer(kind=kint) :: check_inside_y
```

Parameters

meshXYZ Derived Type: ppohDEM_mesh
voxel Derived Type: ppohDEM_voxel
gridco Derived Type: ppohDEM_gridco
grid Derived Type: ppohDEM_vec3
number_mesh: Number of mesh
check_inside_y: Check inside or not for y direction

Uses the following Modules

ppohDEM_util

Called by the following subroutines in ppOpen-APPL/DEM-Util

ppohDEM_v_distanceprecise

Calles the following subroutines

(none)

check_point_inside_out_z

This subroutine determines whether each grid inside the three dimensional objects or not for the z direction.

```
subroutine check_point_inside_out_z (meshXYZ,voxel,gridco,grid,number_mesh,check_inside_z)

    use ppohDEM_util

    type(ppohDEM_mesh) :: meshXYZ
    type(ppohDEM_voxel) :: voxel
    type(ppohDEM_gridco) :: gridco
    type(ppohDEM_vec3) :: grid
    integer(kind=kint) number_mesh
    integer(kind=kint) :: check_inside_z
```

Parameters

meshXYZ Derived Type: ppohDEM_mesh
voxel Derived Type: ppohDEM_voxel
gridco Derived Type: ppohDEM_gridco
grid Derived Type: ppohDEM_vec3
number_mesh: Number of mesh
check_inside_z: Check inside or not for z direction

Uses the following Modules

ppohDEM_util

Called by the following subroutines in ppOpen-APPL/DEM-Util

ppohDEM_v_distanceprecise

Calles the following subroutines

(none)

output_data

This subtoutine outputs the distance data file.

```
subroutine output_data (file_info,voxel)

    use ppohDEM_util

    type(ppohDEM_fileinfo) :: file_info
    type(ppohDEM_voxel) :: voxel
```

Parameters

file_info Derived Type: ppohDEM_fileinfo
voxel Derived Type: ppohDEM_voxel

Uses the following Modules

ppohDEM_util

Called by the following subroutines in ppOpen-APPL/DEM-Util

(none)

Calles the following subroutines

(none)

ppohDEM_v_exit

This subtoutine deallocates variables.

```
subroutine ppohDEM_v_exit (meshXYZ,voxel,gridco)

    use ppohDEM_util

    type(ppohDEM_mesh) :: meshXYZ
    type(ppohDEM_voxel) :: voxel
    type(ppohDEM_gridco) :: gridco
```

Parameters

meshXYZ Derived Type: ppohDEM_mesh
voxel Derived Type: ppohDEM_voxel
gridco Derived Type: ppohDEM_gridco

Uses the following Modules

ppohDEM_util

Called by the following subroutines in ppOpen-APPL/DEM-Util

(none)

Calles the following subroutines

(none)

3.3.2 ppOpen-APPL/DEM

ppohDEM_loadfileinfo

This subroutine reads the input file containing information for ppOpen-APPL/DEM libraries.

```
subroutine ppohDEM_loadfileinfo (file_info)

    use ppohDEM_util

    type(ppohDEM_fileinfo) :: file_info
```

Parameters

file_info Derived Type: ppohDEM_fileinfo

Uses the following Modules

ppohDEM_util

Called by the following subroutines in ppOpen-APPL/DEM

(none)

Calles the following subroutines

(none)

ppohDEM_setparameters

This subroutine reads input files and sets the parameters for ppOpen-APPL/DEM libraries.

```
subroutine ppohDEM_setparameters (file_info,parameters)

        use ppohDEM_util

        type(ppohDEM_fileinfo) :: file_info
        type(ppohDEM_parameters) :: parameters
```

Parameters

file_info Derived Type: ppohDEM_fileinfo
parameters Derived Type: ppohDEM_parameters

Uses the following Modules

ppohDEM_util

Called by the following subroutines in ppOpen-APPL/DEM

(none)

Calles the following subroutines

(none)

ppohDEM_v_pre

This subroutine prepares the variables for ppOpen-APPL/DEM libraries.

```
subroutine ppohDEM_pre (file_info,parameters,particles,cells,walls,objects,counter)
        use ppohDEM_util
        type(ppohDEM_fileinfo) :: file_info
        type(ppohDEM_parameters) :: parameters
        type(ppohDEM_particles) :: particles
        type(ppohDEM_cells) :: cells
        type(ppohDEM_walls) :: walls
        type(ppohDEM_objects) :: objects
        type(ppohDEM_counter) :: counter
```

Parameters

file_info Derived Type: ppohDEM_fileinfo
parameters Derived Type: ppohDEM_parameters
particles Derived Type: ppohDEM_particles
cells Derived Type: ppohDEM_cells
walls Derived Type: ppohDEM_walls
objects Derived Type: ppohDEM_objects
counter Derived Type: ppohDEM_counter

Uses the following Modules

ppohDEM_util

Called by the following subroutines in ppOpen-APPL/DEM

(none)

Calles the following subroutines

ppohDEM_prepare_objects_fromfile

ppohDEM_prepare_objects_fromfile

This subtoutine reads the distance data file and sets the computational space in the distance data space.

```
subroutine ppohDEM_prepare_objects_fromfile(parameters,objects)

    use ppohDEM_util

    type(ppohDEM_parameters) :: parameters
    type(ppohDEM_objects) :: objects
```

Parameters

parameters Derived Type: ppohDEM_parameters
objects Derived Type: ppohDEM_objects

Uses the following Modules

ppohDEM_util

Called by the following subroutines in ppOpen-APPL/DEM

ppohDEM_v_pre

Calles the following subroutines

(none)

ppohDEM_update_index

This subtoutine updates the indexes of particles and cells.

```
subroutine ppohDEM_update_index(parameters,particles,cells)

        use ppohDEM_util

        type(ppohDEM_parameters) :: parameters
        type(ppohDEM_objects) :: objects
        type(ppohDEM_cells) :: cells
```

Parameters

parameters Derived Type: ppohDEM_parameters
objects Derived Type: ppohDEM_objects
cells Derived Type: ppohDEM_cells

Uses the following Modules

ppohDEM_util

Called by the following subroutines in ppOpen-APPL/DEM

(none)

Calles the following subroutines

(none)

ppohDEM_checkparameters

This subroutine checks the parameters.

```
subroutine ppohDEM_checkparameters(parameters,particles,cells)

        use ppohDEM_util

        type(ppohDEM_parameters) :: parameters
        type(ppohDEM_particles) :: particles
        type(ppohDEM_cells) :: cells
```

Parameters

parameters Derived Type: ppohDEM_parameters
particles Derived Type: ppohDEM_particles
cells Derived Type: ppohDEM_cells

Uses the following Modules

ppohDEM_util

Called by the following subroutines in ppOpen-APPL/DEM

(none)

Calles the following subroutines

(none)

ppohDEM_check_update

This subroutine checks and updates the indexes.

```
subroutine ppohDEM_check_update(parameters,particles,cells,counter)

        use ppohDEM_util

        type(ppohDEM_parameters) :: parameters
        type(ppohDEM_particles) :: particles
        type(ppohDEM_cells) :: cells
        type(ppohDEM_counter) :: counter
```

Parameters

parameters Derived Type: ppohDEM_parameters
particles Derived Type: ppohDEM_particles
cells Derived Type: ppohDEM_cells
counter Derived Type: ppohDEM_counter

Uses the following Modules

ppohDEM_util

Called by the following subroutines in ppOpen-APPL/DEM

(none)

Calles the following subroutines

(none)

ppohDEM_integrate_system

This subroutine calls a subroutine for calculating force.

```
subroutine ppohDEM_integrate_system(parameters,particles,cells,walls,objects)

    use ppohDEM_util

    type(ppohDEM_parameters) :: parameters
    type(ppohDEM_particles) :: particles
    type(ppohDEM_cells) :: cells
    type(ppohDEM_walls) :: walls
    type(ppohDEM_objects) :: objects
```

Parameters

parameters Derived Type: ppohDEM_parameters
particles Derived Type: ppohDEM_particles
cells Derived Type: ppohDEM_cells
walls Derived Type: ppohDEM_walls
objects Derived Type: ppohDEM_objects

Uses the following Modules

ppohDEM_util

Called by the following subroutines in ppOpen-APPL/DEM

(none)

Calles the following subroutines

ppohDEM_calcforce

ppohDEM_calcforce

This subroutine calculates force of particles.

```
subroutine ppohDEM_calcforce(parameters,particles,cells,walls,objects)

        use ppohDEM_util

        type(ppohDEM_parameters) :: parameters
        type(ppohDEM_particles) :: particles
        type(ppohDEM_cells) :: cells
        type(ppohDEM_walls) :: walls
        type(ppohDEM_objects) :: objects
```

Parameters

parameters Derived Type: ppohDEM_parameters
particles Derived Type: ppohDEM_particles
cells Derived Type: ppohDEM_cells
walls Derived Type: ppohDEM_walls
objects Derived Type: ppohDEM_objects

Uses the following Modules

ppohDEM_util

Called by the following subroutines in ppOpen-APPL/DEM

ppohDEM_integrate_system

Calles the following subroutines

(none)

ppohDEM_check_objects

This subroutine checks objects and calculates motion of translation and quaternion rotation.

```
subroutine ppohDEM_check_objects(parameters,objects,counter)

        use ppohDEM_util

        type(ppohDEM_parameters) :: parameters
        type(ppohDEM_objects) :: objects
        type(ppohDEM_counter) :: counter
```

Parameters

parameters Derived Type: ppohDEM_parameters
objects Derived Type: ppohDEM_objects
counter Derived Type: ppohDEM_counter

Uses the following Modules

ppohDEM_util

Called by the following subroutines in ppOpen-APPL/DEM

(none)

Calles the following subroutines

ppohDEM_output_object_pov

ppohDEM_output_object_pov

This subroutine outputs pov data files for visualisation.

```
subroutine ppohDEM_output_object_pov(objects,counter)

    use ppohDEM_util

    type(ppohDEM_objects) :: objects
    type(ppohDEM_counter) :: counter
```

Parameters

objects Derived Type: ppohDEM_objects
counter Derived Type: ppohDEM_counter

Uses the following Modules

ppohDEM_util

Called by the following subroutines in ppOpen-APPL/DEM

ppohDEM_check_objects

Calles the following subroutines

(none)

ppohDEM_dataout

This subroutine outputs data files.

```
subroutine ppohDEM_dataout (file_info,parameters,particles,counter)

        use ppohDEM_util

        type(ppohDEM_fileinfo) :: file_info
        type(ppohDEM_parameters) :: parameters
        type(ppohDEM_particles) :: particles
        type(ppohDEM_counter) :: counter
```

Parameters

file_info Derived Type: ppohDEM_fileinfo
parameters Derived Type: ppohDEM_parameters
particles Derived Type: ppohDEM_particles
counter Derived Type: ppohDEM_counter

Uses the following Modules

ppohDEM_util

Called by the following subroutines in ppOpen-APPL/DEM

(none)

Calles the following subroutines

(none)

ppohDEM_exit

This subtoutine deallocates variables.

```
subroutine ppohDEM_exit(particles,cells,walls,objects)

        use ppohDEM_util

        type(ppohDEM_particles) :: particles
        type(ppohDEM_cells) :: cells
        type(ppohDEM_walls) :: walls
        type(ppohDEM_objects) :: objects
```

Parameters

particles Derived Type: ppohDEM_particles
cells Derived Type: ppohDEM_cells
walls Derived Type: ppohDEM_walls
objects Derived Type: ppohDEM_objects

Uses the following Modules

ppohDEM_util

Called by the following subroutines in ppOpen-APPL/DEM

(none)

Calles the following subroutines

(none)