Quickstart guide for the MPD program

Jérémy Dalphin*

April 11, 2019

The MPD program is a software written in C, which aims to compute Maximal Probability Domains (MPDs) as illustrated in Figure 1, *i.e.* regions of the real three-dimensional space that maximize the probability to find exactly ν electrons inside, and $n-\nu$ ones outside, given at least:

- a chemical configuration of n electrons (via a **.wfn* file for Hartree-Fock wave function);
- a prescribed (positive) number $\nu \in \{1, \ldots, n\}$ of electrons to look for;
- and optionally an initial mesh to start with (via a **.mesh* or **.cube* file).

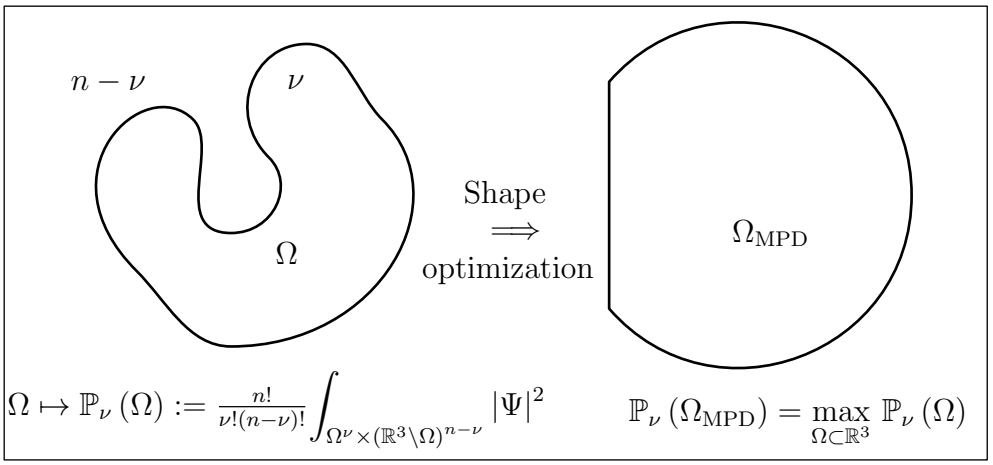

Figure 1: Illustration of the shape optimization process leading to a maximal probability domain.

1 The **.info* format as a single input file

The MPD program only needs one input argument: the name of a **.info* file that contains a specific syntax (an enumeration of keywords and associated values finished with *end_data*). By default, the length of the **.info* file name must be at most 100 characters (no whitespace is allowed).

The MPD program can be installed from *https://github.com/ISCDtoolbox/MPD.git* repository. We recommend to perform tests in local subdirectories of *MPD/outputFiles*, where different **.info* files can be saved. For example, let us work in *MPD/outputFiles/quickstartTests/test1*. Then, the user can launch the MPD program from *quickstart1.info* by typing in a command prompt:

```
../../../bin/mpdProgram quickstart1.info
```

or equivalently by navigating to the *MPD/bin* directory and typing in a command prompt:

```
./mpdProgram ../outputFiles/quickstartTests/test1/quickstart1.info
```

The preprocessing step (iteration -1) initializes a structure called *Parameters* with default values (the preprocessor constants of *loadParameters.h* source file). Then, the **.info* input file is read, and the structure is modified according to the scanned values. Finally, the content of *Parameters* is checked to see if all values are compatible with each other.

* *Jérémy Dalphin (jeremy.dalphin@mines-nancy.org) is a Research Engineer at the Institut des Sciences du Calcul et des Données (ISCD), Sorbonne Université (Université Pierre et Marie Curie), 4 place de Jussieu, boîte courrier 380, 75252 Paris Cedex 5, France.*

1.1 Two typical examples of *.info input files

The MPD program is currently able to represent domains inside a computational box, either as a reunion of small hexahedra inside a given cubical grid; or as a reunion of tetrahedra inside the prescribed mesh of a cube. Consequently, the numerical MPD algorithm is very different if one considers hexahedral meshes (default optimization mode) or tetrahedral ones. We now give two examples of **.info* files, each one corresponding to the type of mesh aforementionned. We recall we are assuming to work in *test0/* or *test1/* subdirectory of *MPD/outputFiles/quickstartTests*.

1.1.1 Example of *test0/quickstart0.info* input file for hexahedral meshes

This example considers the H_2O molecule and tries to maximize the probability to find exactly 2 electrons, starting from a cube of length 2 centred on an H atom. It will use 1 processor and will save the (hexahedral) mesh every 25 iteration in **.mesh* format and in the **.cube* one at each step. It will stop if convergence is achieved (*i.e.* if the domain no longer evolves) or if the total number of steps exceeds 500 iterations. The software *medit* can be launched to read the **.mesh* files.

```
opt_mode 0
n_cpu 1

name_chem ../../../inputFiles/h2o-def2tzvp-rhf.wfn
nu_electrons 2

ls_y 1.43154464
ls_z -0.88753579
ls_r 2.

save_type 2
save_mesh 25

iter_max 500
path_medit ../../../bin/medit

end_data

What is placed after end_data in this file is not read and considered as comments.
The order of the lines before end_data is not important as well as the space between them.
This file is entitled quickstart0.info and can be used as a valid input for the MPD program.
```

1.1.2 Example of *test1/quickstart1.info* input file for tetrahedral meshes

This example considers the H_2O molecule and tries to maximize the probability to find exactly 2 electrons, starting from a unit sphere centred on the O atom. It will use 1 processor and will save the (tetrahedral) mesh in the **.obj* format at each step, and also the initial and last one in the **.mesh* format. It will stop if the variation of probability is less than 5×10^{-5}, if the probability residual is less than 5×10^{-4} and if the variation of probability residual is less than 5×10^{-3}.

```
opt_mode 1
n_cpu 1

name_chem ../../../inputFiles/h2o-def2tzvp-rhf.wfn
nu_electrons 2

ls_type 1
ls_z 0.22188395

save_type 2

iter_told0p 5.e-5
iter_told1p 5.e-4
iter_told2p 5.e-3

path_medit ../../../bin/medit
path_mmg3d ../../../bin/mmg3d
path_mshdist ../../../bin/mshdist
path_elastic ../../../bin/elastic
path_advect ../../../bin/advect

end_data

What is placed after end_data in this file is not read and considered as comments.
The order of the lines before end_data is not important as well as the space between them.
This file is entitled quickstart1.info and can be used as a valid input for the MPD program.
```

We now detail all the different types of parameters that can be specified in the **.info* input file. Apart from the two mandatory ones *nu_electrons* and *name_chem*, they will be set to their default values if they are not prescribed in the **.info* file. We also mention that if all parameters have been successfully loaded (iteration -1 completed), an **.restart* file is generated and it contains the parameters given in the **.info* file and the default values associated with all the remaining ones.

1.2 Mandatory parameters

nu_electrons It is followed by the number of electrons $\nu \in \{1, \ldots, n\}$ the user is looking for, with n the total number of electrons in the chemical system. We recall that the MPD program computes the domain Ω of the three-dimensional real space $\mathbb{R}^3$ that maximizes the probability to find exactly ν electrons in Ω, and $n - \nu$ ones in the complement $\mathbb{R}^3 \backslash \Omega$.

name_chem It is followed by the (relative/absolute) location of the chemical file containing the electronic configuration the user wants to study. The MPD program can currently read the **.wfn* file format (output of *Gaussian* software) for molecules described by a single Slater determinant wave function (restricted, restricted open-shell, or unrestricted Hartree-Fock). Another similar format produced by the MPD program can be read and it has the **.chem* extension.

```
This file is entitled h2-sto6g.wfn and is located in MPD/inputFiles directory, dist 0.74 Angs
GAUSSIAN 1 MOL ORBITALS 12 PRIMITIVES 2 NUCLEI
H 1 (CENTRE 1) 0.00000000 0.00000000 0.69919867 CHARGE = 1.0
H 2 (CENTRE 2) 0.00000000 0.00000000 -0.69919867 CHARGE = 1.0
CENTRE ASSIGNMENTS 1 1 1 1 1 1 2 2 2 2 2 2
TYPE ASSIGNMENTS 1 1 1 1 1 1 1 1 1 1 1 1
EXPONENTS 0.3552322D+02 0.6513144D+01 0.1822143D+01 0.6259553D+00 0.2430767D+00
EXPONENTS 0.1001124D+00 0.3552322D+02 0.6513144D+01 0.1822143D+01 0.6259553D+00
EXPONENTS 0.2430767D+00 0.1001124D+00
MO 1 MO 0.0 OCC NO = 2.0000000 ORB. ENERGY = -0.582889
0.52158668D-01 0.78724076D-01 0.10339790D+00 0.10201067D+00 0.56401418D-01
0.90740316D-02 0.52158668D-01 0.78724076D-01 0.10339790D+00 0.10201067D+00
0.56401418D-01 0.90740316D-02
END DATA
TOTAL ENERGY = -1.125372194464 THE VIRIAL(-V/T)= 1.93058566
The first line and the ones after END DATA are not read and considered as comments.
```

Other examples can be found in *MPD/inputFiles*. The MPD program always produces a local copy (saved in the **.chem* format) of the original **.wfn/*.chem* file specified by *name_chem*. By default, the name is the one of the **.info* file but where the *.info* extension is replaced by *.chem*. For example, *quickstart1.chem* is created in *MPD/outputFiles/quickstartTests/test1* and corresponds, up to the file format, to the electronic configuration given in *MPD/inputFiles/h2-sto6g.wfn*.

```
MolecularOrbitals 2
Primitives 12
Nuclei
2
0.00000000e+00 0.00000000e+00 6.99198670e-01 1
0.00000000e+00 0.00000000e+00 -6.99198670e-01 1
Coefficient Exponent Center Type
1
Spin 1
5.21586680e-02 3.55232200e+01 1 1
...
9.07403160e-03 1.00112400e-01 2 1
2
Spin -1
5.21586680e-02 3.55232200e+01 1 1
...
9.07403160e-03 1.00112400e-01 2 1
End
What is placed after End in this file is not read and considered as comments.
This file is entitled quickstart1.chem and can be used with name_chem in the quickstart1.info file.
```

1.3 Basic parameters

verbose It is followed by 1 or 0 (default value). If set to one, a lot of execution details are printed in the command prompt, otherwise very little is shown during the execution of the MPD program.

n_cpu It is followed by the positive number of threads the user can prescribe for a parallel thus faster computation of the overlap matrix (and renormalization of the level-set function by *mshdist*). One is the default value and **do not** exceed the total number of cpu available on your computer.

opt_mode It is followed by a number (usually one, zero is the default value) that corresponds to the type of optimization used in the MPD algorithm. Positive values concern tetrahedral meshes whereas non-positive values concern hexahedral meshes (cubical grids):

- 1 combines all tetrahedral methods to get a converged MPD domain with a mesh adapted to both its geometry and the molecular orbitals appearing in the Hartree-Fock wave function;
- 0 starts by deforming the pixelated domain according to some shape gradient perturbations, then switches to an exhaustive search when the MPD region oscillates and no longer evolves;
- −1 deforms iteratively the hexahedral domain according to the local sign of the shape gradient computed at the centers of its boundary quadrilaterals (and the closest parallel ones);
- −2 tries to add and/or remove successively the hexahedra touching the boundary quadrilaterals of the domain by keeping only local deformations that increase the resulting probability;
- 2 performs Eulerian deformations of the level-set function (implicitly encoding the tetrahedral domain via its signed distance) by considering a vector-field extension of the shape gradient;
- 3 computes Lagrangian deformations that explicitly deform the tetrahedral mesh according to a vector-field extension of the shape gradient (it should be used for small perturbations only);
- 4 defines iteratively a new tetrahedral domain as the positive part of the previous shape gradient (this very unstable mode should be used occasionally to get the MPD shape far from nuclei).

path_* (medit/mmg3d/mshdist/elastic/advect) It is followed by the (relative/absolute) location of the external softwares launched by the MPD algorithm. The symbol ∼ cannot be used for the home directory in the path names. Since we assume the user works in subdirectories of *MPD/outputFiles*, the default path names are *../../bin/medit*, etc. They can be prescribed only if they are needed: *path_mmg3d* (mesh adaptation) and *path_mshdist* (signed distance generator) can be set only for tetrahedral meshes *i.e.* if *opt_mode* is set to a positive value; *path_elastic* (linear elasticity extension of the shape gradient) and *path_advect* (advection of the level-set function) only if *opt_mode* is set to 1 or 2; *path_medit* (3D mesh visualization) for any optimization mode. We recommend to specify absolute paths if one aims to copy a file from one directory to another.

***_length (path/name)** It is followed by a number greater than its default value (101). It can be used to increase the maximal number of characters allowed with *path_** or *name_** keywords.

Remarks The default value associated with any parameter can be modified in the header file *MPD/sources/mpd/sources/loadParameters.h*. For example, the default path name for mmg3d can be changed accordingly at the line *#define PATH_MMG3D "./../../bin/mmg3d"*. Similarly, maximal length for the **.info* file name can be defined at the line *#define NAME_LENGTH 101*. This should be done before the installation, otherwise the MPD program must be compiled again. We recall that whitespaces should **never** be used for the names of your local directories or files.

1.3.1 Parameters related to the computational box

During the preprocessing step (iteration −1), an initial mesh (file already existing in the **.mesh* or **.cube* format) can be prescribed thanks to the *name_mesh* keyword, otherwise a computational box is built according to the *n_**, *_min*, *_max*, and *delta_** ($x/y/z$) parameters. If nothing is particularly specified, a default computational box $[-3.5, 3.5]\times[-3.5, 3.5]\times[-3.5, 3.5]$ is discretized with $(71)\times(71)\times(71)$ points and $(0.1)\times(0.1)\times(0.1)$ space step precision.

name_mesh It is followed by the (relative/absolute) location of the mesh file name the user wants to start with. The MPD program can currently read the **.cube* and **.mesh* file format.

- If the **.mesh/*.cube* file exists, then it is read and a local copy is made by using the **.info* file name, but where the *.info* extension is replaced by the *.mesh* one. This default file name is also associated with *name_mesh* if nothing has been prescribed.
- If such a file does not exist, then the name is used to save the initial mesh of the computational box, which is a cube [x_min, x_max] × [y_min, y_max] × [z_min, z_max] discretized with (n_x) × (n_y) × (n_z) points and (delta_x) × (delta_y) × (delta_z) space step precision.

***_min (x/y/z)** It is followed by the minimal coordinate of the computational cube in the associated (x/y/z)-direction, which must be lower than *(x/y/z)_max* (−3.5 is the default value).

***_max (x/y/z)** It is followed by the maximal coordinate of the computational cube in the corresponding (x/y/z)-direction, which must be greater than *(x/y/z)_min* (3.5 is the default value).

n_* (x/x/z) It is followed by the number of discretization points in the corresponding (x/y/z)-direction, which must be greater than two (71 is the default value).

delta_* (x/y/z) It is followed by the space discretization step in the corresponding (x/y/z)-direction, which must be positive (0.1 is the default value) and which is given by the relation:

$$\text{delta\_(x/y/z)} = \frac{\text{(x/y/z)\_max} - \text{(x/y/z)\_min}}{\text{n\_(x/y/z)} - 1}.$$

Let us now describe the **.mesh* file format (input of *medit* software, used to visualize meshes). It must begin with *MeshVersionFormatted 2 Dimension 3 Vertices* followed by the total number of mesh points and an enumeration of their three coordinates (double precision) and associated label (integer value). It must finish with the *End* keyword (everything after is considered as comments). In-between, the file is made of successive blocks. Each one starts with the type of elements considered (*Edges*/*Triangles*/*Tetrahedra*/*Quadrilaterals*/*Hexahedra*) followed by their total number in the mesh. Then, an enumeration of their vertices as listed before (10 is the tenth vertex) and associated label. For tetrahedral meshes, one can also specify some geometric informations: *Corners*/*RequiredVertices*/*Ridges* can be used to list the vertices/edges concerned, *Normals*/*Tangents* to define the three coordinates of some normal/tangent vectors, and *NormalAtVertices*/*TangentAtVertices* to successively relate the point and vector considered. An example of *quickstart1.ini.mesh* file can be found in the *MPD/inputFiles* directory and is given thereafter.

The **.cube* output format of *Gaussian* software can also be used with *name_mesh* to specify an initial mesh in the **.info* input file. We recommend to use this input format only for hexahedral meshes *i.e.* for non-positive values of *opt_mode*. First, such a file starts by specifying the number of atoms and the three center coordinates of the lowest hexahedron. In each (x/y/z)-direction, we next prescribe the number of cubes and space discretization step given as a vector. Then, we enumerate each atom description as its charge (integer value followed by the corresponding double one) and its three point coordinates. Finally, we define a collection of labels (double values) associated with the centres of the cubical grid, depending if they belong to the internal domain (1.0) or not (0.0). The following example of *quickstart0.ini.cube* file can be found in *MPD/inputFiles*.

```
This file is entitled quickstart0.ini.cube and can be used with name_mesh in the quickstart0.info file.
It is located in the MPD/inputFiles directory. The first two lines are considered as comments.
3 -3.45000000e+00 -3.45000000e+00 -3.45000000e+00
70 0.100000 0.000000 0.000000
70 0.000000 0.100000 0.000000
70 0.000000 0.000000 0.100000
1 1.000000 0.00000000e+00 1.43154464e+00 -8.87535790e-01
8 8.000000 0.00000000e+00 0.00000000e+00 2.21883950e-01
1 1.000000 0.00000000e+00 -1.43154464e+00 -8.87535790e-01
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
...
0.000000 0.000000 0.000000 0.000000
```

```
MeshVersionFormatted 2
Dimension 3
Vertices
30134
-0.352243444499497 0.29826663478732 -0.214769786718314 0
...
-6.15284263e-02 -6.71025523e-01 1.06546645e+00 0
Corners
8
926
...
6343
RequiredVertices
8
926
...
6343
Tetrahedra
168800
6312 6366 6307 6476 2
...
30134 3083 30057 22720 2
Normals
1031
3.64277197829145e-15 -9.20287130448727e-16 1.00000000e+00
...
1.00000000e+00 6.77944363367812e-16 -8.15936752802458e-16
NormalAtVertices
1031
8 1
...
29939 1031
Tangents
86
1.00000000e+00 -3.86460910350384e-16 -3.86460910350384e-17
...
5.78149661312982e-16 0.00000000e+00 -1.00000000e+00
TangentAtVertices
86
22 1
...
9130 86
Triangles
2242
10720 5730 5237 1
...
1152 29939 1199 2
Edges
98
3852 4085 0
...
6003 6021 0
Ridges
98
1
...
98
End
```

What is placed after the End keyword is considered as comments. The order of the blocks is not important. This file is entitled quickstart1.ini.mesh and can be used with name_mesh in the quickstart1.info file.

1.3.2 Parameters related to the initial domain

During the initialization step (iteration 0), a starting domain is built and the mesh is adapted accordingly inside the computational box. If an existing initial mesh has been prescribed with *name_mesh*, the MPD program tries to detect if an internal domain is not already hidden in its structure (hexahedra/tetrahedra labelled 3 or 2 if they belong or not to the internal domain, quadrilaterals/triangles labelled 10 if they are parts of its boundary). Otherwise, an initial cube/sphere can be prescribed by specifying its three center coordinates and its length/radius thanks to the following parameters. Note also that a numerical half-space is just a well-chosen starting cube.

ls_type It is followed by 1 or 0 (default value). If set to one, the initial domain is a sphere, otherwise it is a cube. We recommend to start with a sphere for tetrahedral meshes and a cube for hexahedral ones. Note also that an initial half-space can be prescribed as a starting domain if the center and length of the initial cube are well-chosen with respect to the computational box.

ls_* (x/y/z) It is followed by the center coordinate of the cube/sphere in the (x/y/z)-direction. By default, the starting domain is centred at the origin (*i.e.* 0.0 is the default value).

ls_r It is followed by the (positive) length/radius of the starting cube/sphere. By default, the starting cube/sphere has unit length/radius (*i.e.* 1.0 is the default value). We emphasize the fact that it specifies the radius (and not the diameter) of a sphere, or the length of a cube.

1.3.3 Parameters related to the stop criteria

The MPD program converges if three parameters are small enough, given a certain tolerance *iter_tol** (*d0p*/*d1p*/*d2p*). A security bound is also imposed on the number of iterations (*iter_max*).

iter_max It is followed by the maximal (non-negative) number of iterations allowed during the optimization process. The default value is 100 and if set to zero, the MPD program stops at the end of initialization (iteration 0). This can be used to compute the probability on a given domain.

iter_told0p It is followed by the (non-negative) tolerance allowed on the variation of probability between two iterations, which is given by $(\Delta\mathbb{P})_0 = \mathbb{P}_0$ and $(\Delta\mathbb{P})_i = \mathbb{P}_i - \mathbb{P}_{i-1}$ for iteration $i > 0$. The default value is $1.0\mathrm{e}-10$ and if set to zero, it is not taken into account in the stop criteria.

iter_told1p It is followed by the (non-negative) tolerance allowed on the probability residual (first-order optimality condition) given by $\int_{\mathrm{bdy}(\Omega_i)}(\frac{\partial\mathbb{P}_i}{\partial\Omega})^2$, with $\frac{\partial\mathbb{P}_i}{\partial\Omega}$ the shape gradient at iter. $i \geqslant 0$. The default value is $1.0\mathrm{e}-10$ and if set to zero, it is not taken into account in the stop criteria.

iter_told2p It is followed by the (non-negative) tolerance allowed either on the second-order variation of probability (hexahedral meshes), or on the variation of the probability residual (tetrahedral meshes) between two iterations. The default value is $1.0\mathrm{e}-10$ and if set to zero, it is not taken into account in the stop criteria. For hexahedra, it is used to detect some domain oscillations.

1.3.4 Parameters related to the saving of data

The numerical values associated with the quantities of interest (probability, population, residual, computational time, ...) are appended in a **.data* file, whose name can be specified (*name_data*) and the frequency prescribed (*save_data*). The meshes can be saved at a given rate (*save_mesh*) in various formats (*save_type*). The *medit* software can be used during the iterative process (*save_print*) to visualize on the mesh some informations (*save_where*): metric, gradient, etc.

name_data It is followed by the (relative/absolute) location of the **.data* file name on which the user wants to append the computed values of interest. By default, if nothing has been prescribed, the **.info* file name is used, but where the *.info* extension is replaced by the *.data* one. Note that such files are always appended (and never deleted or overwritten) by the MPD program at the end of each iteration completed. It allows to keep track of all the tests performed but in return, their size can increase critically after many similar tests. Here is an example of *quickstart1.data* file.

```
Wed Apr 10 12:09:08 2019
Maximizing the probability to find 2 electrons inside the domain and 8 outside.
Iter. Prob. Pop. Dprob. Res. Dres. Time (in min.) Cumulated time (in min.)
0 0.03230939 4.76682448 3.23093902e-02 5.47587570e-03 5.47587570e-03 0.35 0.58
1 0.20024609 0.89766659 1.67936704e-01 4.99170999e+01 4.99116241e+01 3.57 4.15
2 0.34603998 1.18897226 1.45793881e-01 3.50344573e+01 1.48826427e+01 0.90 5.05
3 0.34605787 1.18900239 1.78929684e-05 3.50155819e+01 1.88753351e-02 0.85 5.90
4 0.55721550 1.54832613 2.11157627e-01 1.34550818e+01 2.15605002e+01 0.92 6.82
5 0.73525572 1.99713833 1.78040225e-01 4.67675929e-02 1.34083142e+01 0.93 7.75
6 0.73699874 2.03675694 1.74302232e-03 3.36404160e-03 4.34035513e-02 0.47 8.22
7 0.73711913 2.02295186 1.20384221e-04 3.04486079e-04 3.05955552e-03 2.70 10.92
8 0.73712302 2.02325914 3.89156120e-06 2.27271262e-04 7.72148178e-05 1.97 12.88
```

save_data It is followed by the (non-negative) rate at which data are saved in the file given by *name_data*. The value is 1 by default, *i.e.* data are saved at the end of each iteration in the optimization process. It can also be set to 0 if one does not want to save any data during the test.

save_type It is followed by 0, 1 (default value), or 2. If set to one, then the saving format is the **.mesh* one. If set to zero, then it is either **.obj* or **.cube*, depending if *opt_mode* is set to a positive value (tetrahedral mesh) or not (hexahedral mesh). If set to two, then the mesh is saved in both formats. The **.mesh* files can be visualized with *medit*, which is useful to look precisely into the mesh elements, but does not display the atoms and their interactions. To overcome this problem, the MPD program can save the domains in formats that are readable by *Jmol* or *Paraview* softwares. For hexahedral meshes, it uses **.cube* files whereas for tetrahedral meshes, the domains are saved in the **.obj* format, which only contain the surface mesh of the domain boundary. In particular, **.obj* files cannot be used as input meshes for the MPD program since they only list the vertex coordinates and triangulated faces as follows.

```
v 4.76493493e-01 5.90786319e-01 8.67738262e-01
...
v -1.10647856e-01 -2.44416157e-01 1.18301708e+00

f 1 2 3
...
f 709 535 108
```

save_mesh It is followed by the (non-negative) frequency at which the mesh is saved during the optimization process in the format(s) specified by *save_type*. In addition, we mention that the initial mesh (iteration 0) and the one related to the current iteration are always saved in the **.mesh* format. The default value is 0, meaning that nothing else is done during the algorithm.

For example, *quickstart0.info* contains the line *save_mesh 25* meaning that during the 27-th iteration, three files appear in the *test0/* directory: *quickstart0.0.mesh*, *quickstart0.25.mesh*, and *quickstart0.27.mesh*. The last file is removed as soon as the iteration 27 ends, but if the MPD program fails before, this file can be used with the **.restart* one (change the *.restart* extension with the *.info* one in a new directory) to start again the test.

save_print It is followed by the frequency at which the mesh is displayed with *medit* during the optimization process. The default value is 0, meaning that the mesh is never displayed.

Warning with save_print: before setting *save_print* to a positive value, check before that *medit* software is working properly, especially if you are working on a distant server with *shh* (do not forget to use the *-X* or *-Y* option). Moreover, at each iteration that *medit* is loaded, the algorithm will stop until a normal exit of *medit* is performed. Hence, this parameter has to be handled with care since it can generate infinite waiting loops. Finally, if set to a positive value, the mesh of the computational box (iteration -1) and of the initial domain (iteration 0) will also be displayed and a manual confirmation will be asked before building the initial mesh (if it is not given with *name_mesh*). Note that if it is set to a value greater than *iter_max*, the mesh will never be displayed during the optimization process, but the three previous security checks (the manual confirmation and two initial mesh displays) will still be active.

save_where It is followed by an integer (7 by default) referring to the property the user wants to visualize on the mesh with *save_print*. If the asked property is not used in the optimization mode chosen, it will never be displayed (all $1-7$ values are available if *opt_mode* is set to 1 or 2):

1 The metric of the chemical system (molecular orbitals) is displayed at the mesh vertices.

2 The values of the level-set function are displayed at the mesh vertices.

3 The scalar values of the (shape) gradient are displayed at the boundary points of the domain.

4 The vectorial values (gradient $\times$ normal vector) are displayed at the domain boundary points.

5 The extension of the gradient outside the domain boundary is displayed at the mesh vertices.

6 The values of the new (advected) level-set function is displayed at the mesh vertices.

7 The adapted mesh of the new (advected) domain is displayed.

1.4 Advanced parameters (to be completed more properly)

nu_spin Not coded (spin-dependant probability (search for two electrons 1 up and 1 down). Set to 0 by default and do not change this value.

orb_rhf Used to detect (1 or 0) restricted Harthree-Fock system and simplify computations. The parameter should be automatically set.

1.4.1 Parameters related to the overlap-matrix computation

trick_matrix (Never change) Recompute all the overlap matrix at each step (for hexahedral mesh) or not (0 or 1, the latter being the default value).

approx_mode Not coded (Simplify integral computations to accelerate calculations, 1 or 0 which is the default value).

1.4.2 Parameters related to the metric computation

met_err Positive tolerance allowed on the error between the molecular orbitals and its $P1$-interpolate on the mesh (0.005 is the default value).

met_min Minimal size allowed for the metric computation ($1.0e-5$ by default).

met_max Maximal size allowed for the metric computation (1.0 by default).

1.4.3 Parameters related to the medit software

Explain briefly how medit work + references
https://www.ljll.math.upmc.fr/frey/logiciels/Docmedit.dir/index.html
h F1 F2 m z Z r R B A, etc.

1.4.4 Parameters related to the elastic software

name_elas file name containing the elasticity parameters. If specified, a local copy is made by replacing the *.info* extension by the *.elas* one. If nothing is specified, this default file name is used to create a **.elas* file. This example is entitled *defaultFile.elas* and is located in *MPD/inputFiles* directory. It is the one generated by default by the MPD program. It contains the default Lamé coefficients.

```
Dirichlet
1
10 Triangles f
Lame
2
2 186000. 3400.
3 186000. 3400.
```

1.4.5 Parameters related to the advect software

delta_t The final time at which the *advect* sofwatre stops (1.0 is the default value).

no_cfl Set to 0 or 1 (default value) depending if one wants to add or not a CFL condition on the resolution of the advection equation of the level-set.

Recall that **.txt* file are generated to give the output of the external softwares if they fails.

1.4.6 Parameters related to the mshdist software

Recall that the *n_cpu* parameter is used here to parallelize the *mshdist* software.

n_iter Maximal number of iteration allowed with *mshdist* software. Zero (default value) means it is not taken into account.

residual Tolerance allowed on the residual used as a stop criteria by the *mshdist* software. Zero (default value) means it is not taken into account.

1.4.7 Parameters related to the mmg3d software

Explain the different modes and that our version can deal with both metric and level-set function in *-ls* mode.

hmode_lag In the Lagrangian mode (*-lag* option), it can be set to 0 (default value), 1, or 2. Zero just deform the mesh according to the given vector field. One performs a few remeshing operation after. Two performs a complete remeshing procedure after that.

hmin_* (iso/met/ls/lag) Minimal size allowed for the edges during the adaptative remeshing procedure (0.05 is the default value).

hmax_* (iso/met/ls/lag) Minimal size allowed for the edges during the adaptative remeshing procedure (1.0 is the default value).

hgrad_* (iso/met/ls/lag) Maximal ratio allowed between two connecting edges during the adaptative remeshing procedure (2.0 is the default value).

hausd_* (iso/met/ls/lag) Tolerance allowed on the error between the ideal surface model and the mesh during the adaptative remeshing procedure (0.01 is the default value).