McCad-Tutorial

This is a tutorial for the CAD interface program McCad. To use the CAD file examples, change directory to …/McCad-Dir/tutorial (=TUTORIAL).

Model Preparation

Surface Types

In the current version of McCad (0.3.0) the surface types that may be used in the CAD model are restricted to the following:

- Planes
- Cylinders
- Cones
- Spheres
- Circular Tori

To check whether the model contains invalid surfaces run McCad with the **-s** option.

You can run this example yourself in subdirectory *TUTORIAL/preparation*.

If the model contains surface types which are not supported by McCad it will be denoted in the surface check summary

McCad -s example1.stp

Surface Information for Solid 1
==========
Number Of Surfaces In Model : 1417

Planes : 421
Cylinder : 746
Cones : 218
Spheres : 0
Toroidal Surfaces : 24

--- invalid surface types ---
Bezier Surfaces : 0
BSpline Surfaces : 4
Surfaces Of Revolution : 0
Surfaces Of Extrusion : 4
Offset Surfaces : 0
Unknown Type Of Surface : 0
==========

Additionally the file **invalidSurfaces.stp** will be created, which contains all faces of the current model that cannot be processed by McCad.

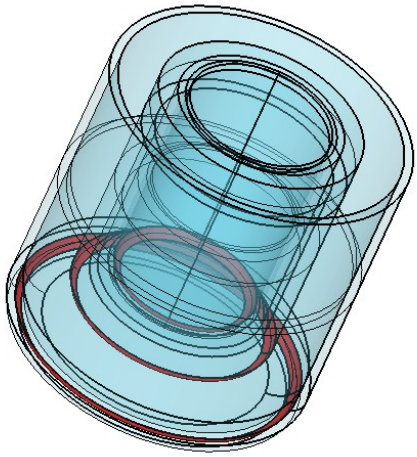

Image 1: The model (translucent) und it's faces of unsupported surface type (red)

To visualize the geometry and it's faces of unsupported surface type open the McCad graphical user interface (call McCad without any parameter) and import the geometry files and the file invalidSurfaces.stp.

To check all STEP files in a directory call **McCad -s …/directory LogFile.out**
A surface summary for all input files in the directory will be saved in LogFile.out.

McCad does not provide tools to substitute unsupported surface types by supported ones. The user needs to take advantage of a CAD system.

Decomposition

For larger models it is advisable to have different directories for the data (material list, bounding box,...), the original model (separated into several subdirectories) and for the processing. A suggestion is given in *TUTORIAL/decomposition*. See Img. 2 as well.

Further it is advisable to store the input geometry in several subdirectories. Each directory in the model directory corresponds to a functional group, e.g. the torodial field coil of a Tokamak fusion device. See Img. 2.
In the subdirectories all solids with a specific material composition can be saved in one single file. This is advisable, since McCad's material management links a STEP file name to a material number and a density.

To decompose a geometry run **McCad -d** on a STEP file or call McCadConvertscript to call McCad on all step files in a directory.
Example two is a simple polyhedral solid (brick.stp) in the directory *TUTORIAL/decomposition/model*.

Set a symbolic link or copy it into directory *conversion*. In the conversion directory call.

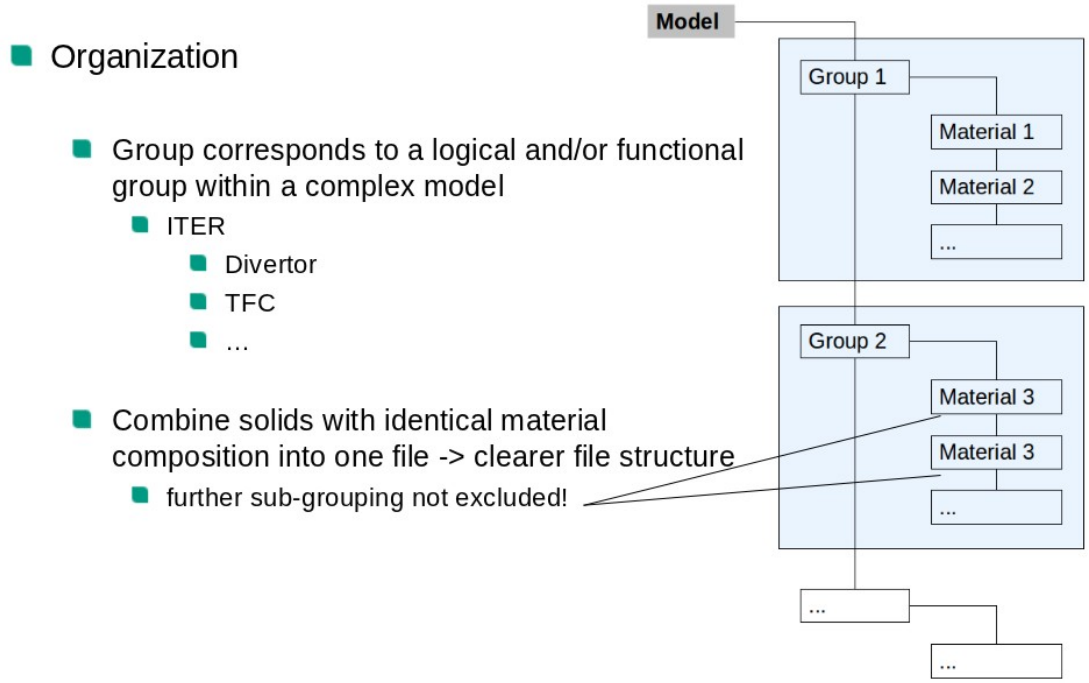

Image 2: Suggestion for the organization of the geometry files of a large model

McCad -d brick.stp

This parameter makes McCad decompose the input solid. If no errors occur the program will generate the file *convertedINPUTFILENAME.stp* which contains the decomposed input geometry still in STEP format.
After the decomposition took place McCad performs a volume comparison between the input geometry and the converted geometry. If the overall volumes differ it will be noted in the file **failedVolumeControl.log** which will not be created if no volume mismatch occurs.

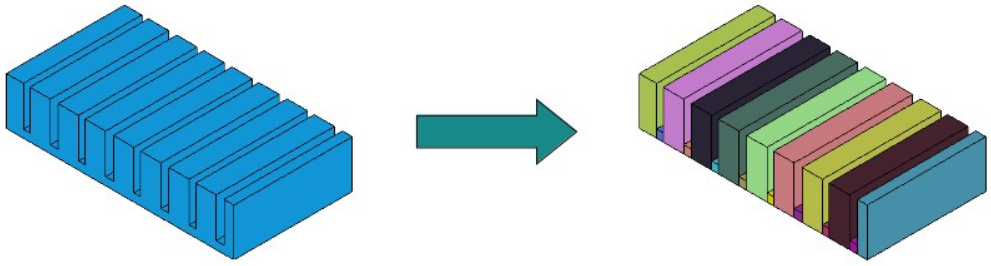

Image 3: Decomposition of a simple geometry

Image 3 shows the original input solid of brick.stp on the left hand side and the decomposed (sign-constant) solid on the right.

It might occur that the model could not be decomposed by McCad. To figure out, what the problem with the model is, it is best to explode a failed file into it's components, i.e. write each solid of a failed

CAD file into a separate file. Change directory to *TUTORIAL/problematic*.
There you'll find a file that fails the decomposition step (problemgeometry.stp).

Image 4: One of these solids does not decompose correctly in McCad

Running **McCad -d problemgeometry.stp** will generate the file failedVolumeControl.log, indicating that the decomposition was not successful.

The file contains two solids from which one fails to decompose. In order to find the problem with the geometry we explode the model into several STEP files from which each contains only one solid.

McCad -e problemgeometry.stp

This results in multiple files (in this case 2) with the naming *ExOut_INPUTFILENAME_NUM.stp* (here ExOut_problemgeometry_NUM.stp) where NUM is a consecutive number for all solids in the inputfile.

Explode a failed file into a separate directory and run the McCadConvertScript, to call McCad -d on all STEP files in the current working directory.

McCadConvertScript .

The script will print a summary of the files that failed completely, i.e. no output file was created. Check also the file failedVolumeControl.log for files that generated output geometry but failed the volume comparison.

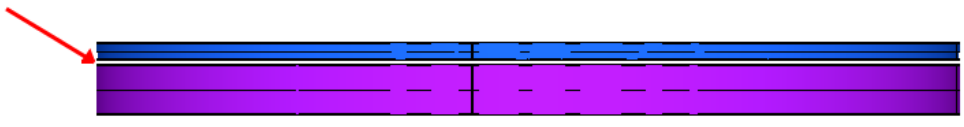

Image 5: Gap after decomposition of solid two (red arrow)

In the case of problemgeometry.stp failedVolumeControl.log indicates that the second solid (purple solid in Img. 4) did not decompose correctly.

Image 5 shows the geometry of file convertedExOut_problemgeometry_2.stp. The red arrow points to a gap that was not part of the input geometry. In such case one has to open the failed file in a CAD system and perform a simplification of the input geometry.

Void Generation

The completion of the model with void space and the output into a geometry description for one of the supported Monte Carlo codes (MC codes supported in this version of McCad are MCNP 5 and Tripoli-4) can be invoked by the parameters -m (MCNP) and -t (Tripoli). In the following example we'll generate the geometry description for MCNP of the brick.stp. Return to the directory *TUTORIAL/decomposition/conversion.*

The void completion can be steered by different parameters which will be defined in an parameter file (here called McCadInputFile.txt). The minimum parameterfile is a list of directories where the program can search for converted*.stp files. Notice that only STEP files beginning with the prefix 'converted' will be concidered as input geometry files.

Before we discuss all optional steering parameters we will produce our first MCNP geometry.

McCad -m McCadInputFile.txt

The completion algorithm leaves three new files in the working directory.

- **Mcnp_Model.mcn**, the geometry description in MCNP syntax
- **CadVolumes**, a list of all cells in the output file and their corresponding volume as calculeted in the CAD system
- **CollisionFile**, a list of all collisions among the input solids

Furthermore a directory named **voxels** is created where the discretized CAD solids of the void geometry is stored.

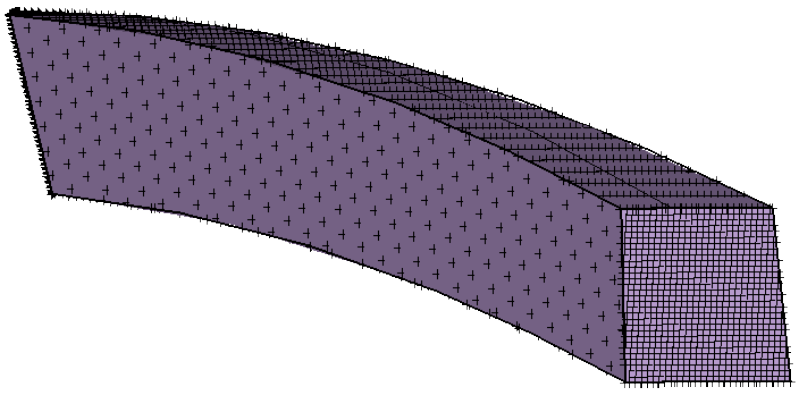

Image 6: Voxel points shown as black crosses

Additional control parameters

The parameter file for the void completion process may contain the following parameters and their default values.

- **InitSurfNb** 1 (integer)
  - initial value for surface numbering
- **InitCellNb** 1 (integer)
  - initial value for cell numbering
- **WriteCollisionFile** 1 (boolean)
  - write CollisionFile of input solids
  a collision test among input solids is performed. If this value is set to '0' or 'false' the program looks for the CollisionFile to read the collisions detected in a previous run. The overlap test is time consuming. In order to save time in a following run with the same geometry but different parameter setting one can disable the WriteCollisionFile parameter.
- **WriteDiscreteModel** 1 (boolean)
  - Write *.voxel files for input solids
  For overlap testing each input solid is covered with sample points, i.e. for each face a mesh of points lying on the face is computed. A voxel file contains the location of all sample points in 3D space. In order to save computation time in a following run with identical geometry set this value to '0' or 'false'. In this case McCad will read the voxel files that have been calculated previously.
- **MinimumInputSolidVolume** 0.1 (double)
  - all input solids with smaller volume will be neglected
  Small volumes of input solids may be negligible for neutronics calculations. This value defines a lower threshold for the volumes of the input solids. It does not directly affect the size of void volumes.
- **MinimumVoidVolume** 1.0 (double)
  - make sure void volumes don't become to small
  This value defines the minimum volume content of void cells.
- **MinimumSizeOfDecompositionFaceArea** 25.0 (double)
  - first cut face area
  For the void generation McCad uses a bounding box which fits the whole input model and decomposes this box using all planar faces and resultants of the input model. This value defines the lower threshold for the face area of a cutting face. All faces with a smaller area will be neglected. This is a parameter to regulate the size and the amount of void cells.
- **MaximumNumberOfComplementedCells** 10 (integer)
  - how many complement operators may be used per void volume
  The void cells are generated by using only planar cut faces for the decomposition of the bounding box. If the model contains non linear faces there will be overlaps between void volumes and solid volumes. The final void cell is generated by using the complement operator. This value defines the maximum number of complement operators used for one void volume (each collision requires one complement operator). It is advisable to keep the number of complement operations low.

- **MaximumNumberOfPreDecompositionCells** 5000 (integer)
  - sets upper threshold for number of void volumes after first cut
  If the number of volumes of the decomposed bounding box exceeds this value before all cut faces of the input geometry has been used for decomposition, the decomposition stops and the overlap test between input solids and void volumes is performed.
- **MinimumSizeOfRedecompositionFaceArea** 1.0 (double)
  - recut if maximum number of complement cells is exceeded
  A void that collides with more input solids than MaximumNumberOfComplementedCells is called here a monster void. For each monster void a second cutting round is performed using all planar faces of the colliding solids bigger than this value.
- **MinimumNumberOfSamplePoints** 10 (integer)
  - lower threshold for number of discretization points along a face's edge
  For the voxelization a mesh of sample voxel points is spread on all faces. These voxel points are used for overlap tests. The higher this value, the less likely it is that an overlap stays undetected. But an increase of this value also means a significant increase in computation time and memory usage.

- **MaximumNumberOfSamplePoints** 50 (integer)
  - upper threshold for number of discretization points along a face's edge
  see MinimumNumberOfSamplePoints above
- **XResolution** 1.0 (double)
  - desired resolution of discretization in x direction
  defines the fineness of the voxelization mesh in combination with the NumberOfSamplePoints values. The finer the mesh the more accurate the overlap test and the more resources are consumed.
- **YResolution** 1.0 (double)
  - desired resolution of discretization in y direction
  see XResolution above
- **Tolerance** 1e-7 (double)
  - tolerance for all geometry related operations – the default value should be sufficient
- **BoundingBox** /path/to/boxfile.stp (string)
  - this string points to a STEP file that contains a bounding box for the input geometry
  the bounding box should be fitted tightly around the model. If no bounding box is defined McCad will create an axis-parallel bounding box. It is favorable to provide a well suited bounding box from a CAD system.
- **MDFile** /path/to/MaterialDensityList.txt (string)
  - path of the file that contains the material information for the problem (MCNP only)
  A MDFile contains a list that links the file names of the input geometry files to a material number and a density (#name #matNb #density)
- **/this/is/a/path/to/converted/files/**
  every line that is no comment or begins with a key word is considered a path to a directory that contains converted*.stp files. Keep in mind that only STEP files with the prefix 'converted' are read by the void generator.

In subdirectory *TUTORIAL/decomposition* edit the McCadInputFile.txt and remove the leading '#' from the line

MDFile ../data/MaterialList.txt

Return to the command line and rerun the McCad. Try also to disable the write parameters for the CollisionFile and the DiscreteModel and see how it improves the conversion speed.

McCad -m McCadInputFile.txt

Take a look at the resulting MCNP file Mcnp_Model.mcn. As you can see the geometry now has material information.

```
McCad generated Input
c          ------ Cells  -------
c          ------ Material cells  ---------- 17
c          ------ Void cells  -------------- 8
c          ------ Outer Void cells  -------- 2
c     McCad: ./ :    brick
1     0
                -1      -2      -3       4       5       6
              IMP:N=1 IMP:P=1
2     0
                 4      -7       1      -2       5      -8
              IMP:N=1 IMP:P=1
3     0
                -2      -7      -6       4       5       9
              IMP:N=1 IMP:P=1
4     0
                 8       4      -3      -2       5     -10
              IMP:N=1 IMP:P=1
5     0
                -9      -2      -3       4       5      11
              IMP:N=1 IMP:P=1
6     0
                 4      -7      10      -2       5     -12
              IMP:N=1 IMP:P=1
7     0
                -2      -7     -11       4       5      13
              IMP:N=1 IMP:P=1
```

```
McCad generated Input
c          ------ Cells  -------
c          ------ Material cells  ---------- 17
c          ------ Void cells  -------------- 8
c          ------ Outer Void cells  -------- 2
c     McCad: ./ :    brick
1     1   -8.01
                -1      -2      -3       4       5       6
              IMP:N=1 IMP:P=1
2     1   -8.01
                 4      -7       1      -2       5      -8
              IMP:N=1 IMP:P=1
3     1   -8.01
                -2      -7      -6       4       5       9
              IMP:N=1 IMP:P=1
4     1   -8.01
                 8       4      -3      -2       5     -10
              IMP:N=1 IMP:P=1
5     1   -8.01
                -9      -2      -3       4       5      11
              IMP:N=1 IMP:P=1
6     1   -8.01
                 4      -7      10      -2       5     -12
              IMP:N=1 IMP:P=1
7     1   -8.01
                -2      -7     -11       4       5      13
              IMP:N=1 IMP:P=1
```

Image 7: Mcnp geometry description without (left) and with (right) material information.

To get a feel for the parameters that coordinate the void generation (MinimumVoidVolume, MinimumSizeOfDecompositionFaceArea, MaximumNumberOfPreDecompositionCells, MinimumSizeOfRedecompositionFaceArea, MaximumNumberOfComplementedCells) play with these parameters for the brick example. For example try to generate just one void cell that excludes all solids it collides with.

Like this

```
c    McCad:  =====  Void Cells  =====
c
18   0
              20    -23     4     -2      5     -7
              #1     #2     #3     #4     #5     #6     #7     #8
              #9    #10    #11    #12    #13    #14    #15    #16
             #17
           IMP:N=1 IMP:P=1
```