
# McCAD v1.0

*Moataz Harb*
*moataz.harb@kit.edu*

*Karlsruhe Institute of Technology (KIT),*
*Hermann-von-Helmholtz-Platz 1, 76344*
*Eggenstein-Leopoldshafen, Germany*

January 12, 2023

# Contents

# 1 Introduction

McCAD is a library with an executable for the conversion of CAD solid models to the input syntax of the Monte Carlo (MC) code MCNP; from Boundary Representation "BREP" to Constructive Solid Geometry "CSG". The library is written in C++ and uses CMake as a build system. It has Open CASCADE Technology and Boost C++ libraries as dependencies. Currently, the library is capable of processing solids with the following surface types - and their combinations:

- planar
- cylindrical
- toroidal
- conical

The general capabilities/output of the library include:

- producing a list of volumes of CAD solids as loaded from the input STEP file(s).
- decomposing complex solids into their constituent convex primitives.
- converting decomposed solids into the input syntax of the MC code MCNP.
- automatic void generation.
- generating entries - surface source and cell tallies - for stochastic cell volume calculation on the generated MC input file.
- producing a mapping of cell IDs on the generated MC input file to the corresponding volumes and names of CAD solids as loaded from the decomposed STEP file(s).
- producing a mapping of void to material cells in the generated MC input file.

# 2 Installation from Source

Building of the McCAD library is supported on both Linux and Windows operating systems (OS). The library has three 3rd-party dependencies: CMake, Boost C++ libraries, and Open CASCADE Technology. CMake is the standard build system for McCAD and it comes by default with most Linux distributions as well as Windows OS. Boost C++ libraries consist of header files that are utilized for parallel processing in McCAD. Open CASCADE Technology (OCCT) is used as a geometry engine for geometrical solids manipulation and decomposition. Below are guiding steps for installation on both Linux and Windows OSs.

## 2.1 Linux

Listed below are the currently supported Linux distributions:

- Ubuntu 20.04 LTS
- Ubuntu 18.04 LTS

Testing of installation on other distributions is still underway! thus Ubuntu is recommended as a distribution to install McCAD. Below are general steps to install McCAD and its dependencies.

- **CMake**
  - download cmake-3.23.0.tar.gz from https://cmake.org/download/ then execute the commands below in a terminal.
  - $ tar -xzvf cmake-3.23.0.tar.gz
  - $ cd cmake-3.23.0
  - $ mkdir build
  - $ cd build
  - $ cmake .. -DCMAKE_USE_OPENSSL=OFF -DCMAKE_INSTALL_PREFIX=.
  - $ make
  - $ make install
- **Boost C++ libraries**
  - download boost_1_78_0.tar.gz from https://www.boost.org/users/download/ then execute the commands below in a terminal.

- $ tar -xvzf boost_1_78_0.tar.gz
- *NOTE*: If Boost libraries are not needed for other applications, then you can skip the remaining steps!
- $ cd boost_1_78_0
- $ mkdir build
- $ cd tools/build
- $ ./bootstrap.sh
- $ ./b2 install --prefix=../../build/

- **Open CASCADE Technology (OCCT)**
  - *NOTE*: the instructions on the installation of dependencies can be found on the side menu in https://dev.opencascade.org/doc/occt-7.5.0/overview/html/index.html by navigating to "Build, Debug and Upgrade > Build 3rd-parties" then following the instructions under "Installation from Official Repositories".
  - download opencascade-7.5.0.tgz from https://dev.opencascade.org/release/previous then execute the commands below in a terminal.
  - $ tar -xzvf opencascade-7.5.0.tgz
  - $ cd opencascade-7.5.0
  - $ mkdir build
  - $ cd build
  - $ cmake .. -DCMAKE_BUILD_TYPE=Release
    -DBUILD_LIBRARY_TYPE=Shared
    -DCMAKE_INSTALL_PREFIX=.
    -DINSTALL_TEST_CASES=true
    -DINSTALL_DOC_Overview=true
  - $ make
  - $ make install

- **McCAD**
  - *NOTE*: building a shared library is recommended! Should a static library be needed, the user has to ensure a compliant build of Open CASCADE Technology by changing "Shared" to "Static" in the build type; -DBUILD_LIBRARY_TYPE=STATIC.

- $ git clone https://github.com/inr-kit/McCAD_Library
- $ cd McCAD_Library
- $ mkdir build
- $ cd build
- to build a shared library (*recommended*):
  - $ CMake ..
    -DCMAKE_INSTALL_PREFIX=.
    -DBUILD_RPATH=true
    -DBUILD_SHARED=true
    -DBUILD_SHARED_EXE=true
    -DBOOST_CUSTOM_ROOT=<Path to boost_1_78_0>
    -DOCC_CUSTOM_ROOT=<Path to opencascade-7.5.0/build>
    -DCMAKE_EXE_LINKER_FLAGS="-m64 -fPIC"
    -DCMAKE_CXX_FLAGS="-fPIC"
- to building a static library:
  - replace "SHARED" by "STATIC" in "
    -DBUILD_SHARED
    -DBUILD_SHARED_EXE in the above commands.
- $ make
- $ make install
- add the absolute path to McCAD/build/bin to .bashrc file or create an alias for McCAD to run from command line.

## 2.2 Windows

Listed below are the currently supported Windows versions (testing of installation on other versions is still underway!):

- Windows 10

Below are the general steps to build McCAD library and its dependencies.

- **CMake** (optional)
  - *NOTE*: If usage of IDE - such Microsoft Visual Studio (VS) - is intended, then installing CMake can be skipped since most IDE builds CMake by default. However, it is recommended to update CMake and adding the full path to the bin folder to the "Path" environment variable.

  - download and run the installer cmake-3.23.1-windows-x86_64.msi from https://cmake.org/download/.

- **Microsoft Visual Studio**
  - download the "community" installer (version 17 2022) from https://visualstudio.microsoft.com/downloads/.
  - run the installer and make sure to enable "Desktop development with C++" on the main menu and ""Windows 10 SDK" on the side menu under "Desktop development with C++".

- **Boost C++ libraries**
  - download boost_1_78_0.zip from https://www.boost.org/users/download/.
  - unzip boost_1_78_0.zip.
  - *NOTE*: if boost will be used for other applications then you can check the documentation in index.html in the unzipped folder for instruction on building the library. Otherwise, no need to build the library since McCAD only uses the header files.

- **Open CASCADE Technology (OCCT)**
  - download and run the installer opencascade-7.5.0-vc14-64.exe from https://dev.opencascade.org/release/previous.
  - after successful installation there will be different folders for the many toolkits that come with a full installation of opencascade; such as freeimage, ffmpeg, openvr, vtk, etc. It is recommended to add the full path to all bin folders to the Windows "Path" environment variable.

- **McCAD**
  - It is recommended to restart the PC/Laptop before proceeding to allow the changes to the "Path" environment from above to take effect.
  - download, by selecting "Code > Download ZIP", source code from https://github.com/inr-kit/McCAD-Library.
  - unzip McCAD_Library.
  - open MSVC and select the McCAD_Library folder.

- from the "Solution Explorer - Folder Review" double click CMake-Settings.json file. This will open the file in IDE.
- Set a "Configuration name".
- ensure that "Configuration type" is set to "Release" and "Toolset" is set to "msvc_x64_x64".
- to build a static library (*recommended*):
  - ensure that the following is on "CMake command arguments"
    -DLINUX_OS=false
    -DBUILD_RPATH=true
    -DBUILD_STATIC=true
    -DBUILD_STATIC_EXE=true
    -DBOOST_CUSTOM_ROOT="<Path to boost_1_78_0>"
    -DOCC_CUSTOM_ROOT="<Path to OpenCASCADE-7.5.0-vc14-64\opencascade-7.5.0>"
- to build a shared library:
  - replace the word "STATIC" by "SHARED" in
    -DBUILD_STATIC
    -DBUILD_STATIC_EXE in the above commands.
- from the top menu select "Build > Build All".
- after successful completion of the previous step, from the top menu select "Build > Install McCAD".
- add the full path to McCAD_Library/build/bin to the PATH environment variable.

# 3 I/O

In general, there are three types of information that need to be provided to McCAD: geometry model(s), materials assignment, and running parameters. Below, the input pertinent to the decomposition and conversion will be highlighted. The output from McCAD will depend on the mode of execution, as will be described in the following sections.

## 3.1 General

The first input to be provided is the input configuration file, McCADInput-Config.i. When run via a command line *without arguments*, McCAD will generate a template configuration file, figure (1), with the default values of

the run parameters. In both run modes, decomposition and conversion, McCAD input parameters are needed. The run parameters are assigned values by default and can be modified by the user. Below is a description of the general input parameters.

```
# McCAD v1.0 run parameters / Thu Jan 12 13:35:30 2023
# ========================

# Input
# =====
# > Debugging level: 0, 1, 2, 3. [0] provides no debugging outputs;
debugLevel = 0
# > The unit used in the input STEP file(s). Only a single unit for all input STEP files!
units = cm
# > Name(s) of the input STEP file(s) separated by a space;
inputFileName = input.stp

# Decomposition
# =============
decompose = true
recurrenceDepth = 20
minSolidVolume = 1.0e-3 [cm3]
minFaceArea = 1.0e-4 [cm2]
scalingFactor = 100.0
precision = 1.0e-6
faceTolerance = 1.0e-8 [cm]
edgeTolerance = 1.0e-8 [cm]
parameterTolerance = 1.0e-8 [cm]
angularTolerance = 1.0e-4 [radian/PI]
distanceTolerance = 1.0e-6 [cm]
simplifyTori = false
simplifyAllTori = false
torusSplitAngle = 30.0 [degrees]

# Conversion
# ==========
convert = false
# > Choose whether to generate void cells;
voidGeneration = false
# > Condition to treat a compound from the STEP file as a single cell, resulting in a unsion of subsolids cells expresseions;
compoundIsSingleCell = false
# > Minimum acceptable void volume should not be less than minSolidVolume;
minVoidVolume = 1.0 [cm3]
# > A larger number will result in fewer void cells but longer cell expressions;
maxSolidsPerVoidCell = 20
# > Choose whether to generate Bound Volume Heirarchy void cells;
BVHVoid = false
# > Choose the desired MC code for conversion;
MCcode = mcnp
startCellNum = 1
startSurfNum = 1
startMatNum = 1
maxLineWidth = 80
# Name of the MC input file to be generated;
MCFileName = MCFile.i
# Name of the Cell ID to solid volume mapping text file to be generated;
volumesFileName = volumes.i
# Name of the void to material cell IDs mapping text file to be generated;
voidCellsFileName = voidCells.i
```

Figure 1: McCAD Input Configuration File

- **debugLevel**: the level at which run/debug info is displayed on the screen. *Default* is 0.
- **units**: the units used in the CAD model. *Default* is cm.
- **inputFileName**: the input STEP file(s) name(s). *Default* is input.stp.

## 3.2 Decomposition

Currently McCAD supports input files in the STEP format (protocol 214). McCAD accepts both single and multiple input STEP files. For each input STEP file, after successfully running McCAD decomposition two STEP output files will be written to disk. The first will contain the decomposed input solids and the second will contain the rejected/failed input solids. If the latter is absent, this means no input solids were rejected and no solids failed to be decomposed.

Before running McCAD in the decomposition mode, the user is advised to review the input parameters on the decomposition section on the input configuration file, figure (1). Below is a description of the decomposition input parameters.

- **decompose**: a logical condition for running decomposition on the input file(s). *Default* is true.
- **recurrenceDepth**: defines the level of decomposition performed recursively on input solids. This input specifies the maximum number of recursive application of decomposition on the subsolids. The larger the number, the less the failed solids. *Default* is 20.
- **minSolidVolume**: controls the recursive splitting of the solids along with the recurrenceDepth. It is also used to filter out the input solids by rejecting solids with smaller volumes than the specified value. *Default* is $10^{-3}$ $cm^3$.
- **minFaceArea**: used to filter out the input solids by rejecting solids with faces smaller in area than the specified value. *Default* is $10^{-4}$ $cm^2$.
- **scalingFactor**: controls the size of generated mesh on solid faces. This parameter mainly affects the collision detection of surfaces. More details can be found in section 8. *Default* is 100.0.
- **precision**: used to judge the equality of numerical values. *Default* is $10^{-6}$.
- **faceTolerance**: more details can be found in section 8. *Default* is $10^{-8}$ cm.
- **edgeTolerance**: more details can be found in section 8. *Default* is $10^{-8}$ cm.

- **parameterTolerance**: more details can be found in section 8. *Default* is $10^{-8}$ cm.
- **angularTolerance**: more details can be found in section 8. *Default* is $10^{-4}$ radian/PI.
- **distanceTolerance**: more details can be found in section 8. *Default* is $10^{-6}$ cm.
- **simplifyTori**: a logical condition on whether to simplify toroidal solids as a collection of smaller cylindrical segments. If *true*, only off axis tori will be simplified. *Default* is false.
- **simplifyAllTori**: a logical condition on whether to simplify all toroidal solids as a collection of smaller cylindrical segments. If *true*, all toroidal solids, disregarding its axis, will be simplified. *Default* is false.
- **torusSplitAngle**: the angle between the top and bottom bases of a cylindrical segment. The angle controls the number of cylindrical segments the torus is simplified to. *Default* is 30.0 degrees.

## 3.3 Conversion

McCAD conversion will result in three textual output files. The first contains the generated MC, MCNP is only support for now, input file. The second contains a mapping of cell ID, volume, and solid name. The third file contains a mapping of void-to-material cell IDs. More details on the format of the three files can be found in section 5.

Material assignment is currently deduced from the name of the input STEP file(s). The user is advised to split the CAD model into several by material. The name of the input STEP files takes the format <material name>_<density in exponential format: -0.0E0>_<.stp. The material name and density are then extracted from the file name and assigned to the solids it contains. For MCNP, the sign of the density implies whether mass or atomic densities is used. In case of absence of the density value, void is assumed. It is important to note that the underscore occurs once in the file name and only separates the material name from density. Any extra underscores will result in an error.

Before running McCAD in the conversion mode, the user is advised to review the input parameters on the conversion section on the input configuration file, figure (1). Below is a description of the conversion input parameters.

- **convert**: a logical condition for running conversion on the input file(s). *Default* is false.
- **voidGeneration**: a logical condition for void generation. If false, only a single void cell is created outside which is a graveyard. *Default* is false.
- **compoundIsSingleCell**: controls whether to define the subsolids of the decomposed input solid into separate cells or unify it through the union operator into a single cell. If *true*, all subsolids are joined through a union operator and a single cell is written to the MC input file. This is not recommended for complex solids since it slows down the transport due to the many surface crossing checks. *Default* is false.
- **minVoidVolume**: controls the desirable minimum void cell volume. *Default* is 1.0 $cm^3$.
- **maxSolidsPerVoidCell**: controls the number of generated void cells. A smaller number will result in a larger number of void cells each with simpler expression on the MC file. *Default* is 40.
- **BVHVoid**: a logical condition for generating a bound volume hierarchy void cells. This feature is still undergoing testing and not currently recommended to be used. *Default* is false.
- **MCcode**: specifies the targeted MC code to generate the input file in a compliant format. Currently only MCNP is supported. *Default* is mcnp.
- **startCellNum**: the desired starting cell number. *Default* is 1.
- **startSurfNum**: the desired starting surface number. *Default* is 1.
- **startMatNum**: the desired starting material number. *Default* is 1.
- **maxLineWidth**: maximum length of lines on the MC file. *Default* is 80.
- **MCFileName**: the desired name for the MC file. *Default* is MCFile.i.
- **volumesFileName**: the desired name for the cell ID-volumes-name mapping file. *Default* is volumes.i.
- **voidCellsFileName**: the desired name of the void to material cells mapping file. *Default* is voidCells.i.

# 4 Running McCAD

There are two ways to run McCAD, via a command line or the SpaceClaim plugin. McCAD can be run from a command line either in a Linux terminal or a Windows command prompt.

## 4.1 Command Line

On a Linux terminal or Windows command prompt McCAD can be run by typing in the path to the executable followed by the desired argument. A good practice is to add the full path of the McCAD executable to the PATH environment variable then simply call "McCAD" with the acceptable desired command line arguments. The arguments could be displayed on the screen by running "McCAD help". The acceptable McCAD arguments are listed below.

- []: running McCAD without arguments will result in writing McCAD template input configuration file, McCADInputConfig.i.
- [**help**]: lists all the acceptable arguments.
- [**read**]: test loading the solids from the input STEP file(s). This is used to test if the OCCT reader will fail to load any STEP file(s). McCAD will generate a text file per input STEP file that contains the volumes of shapes contained in the file.
- [**run**]: executes McCAD according to the selected running mode on the McCADInputConfig.i file; decomposition, conversion, or both.

## 4.2 SpaceClaim Plugin

The python script located in plugins/SpaceClaim_interface can be loaded into SpaceClaim and used directly to call McCAD to perform decomposition on selected solid(s). The script can be loaded in SpaceClaim by selecting "File > New > Script". Select the desired solid(s) to decompose and on the "Script Editor" select "Run Script", the green arrow.

Figure (2) shows a screen shot from a loaded script in SpaceClaim. After loading the script into SpaceClaim, the user should modify *McCADExecutable* and *workPath* variables according. The former should be modifyied to point to McCAD executable that will be called by the script. The latter is the desired work directory that the temporary files will be created in. The

user then can select solid(s) from the model and run the plugin using the green arrow at the top right corner. After running McCAD, a summary will be printed in the "Interpreter" as shown at the bottom of figure (2).

If the user which to modify the input configuration file, McCADInputConfig.i, there are two way to do so. The first is to copy the template from the templates folder and place it in the "workPath" folder. Another way is by running the script without selecting any solids. This way McCAD will generate the McCADInputConfig.i file which the user can modify.

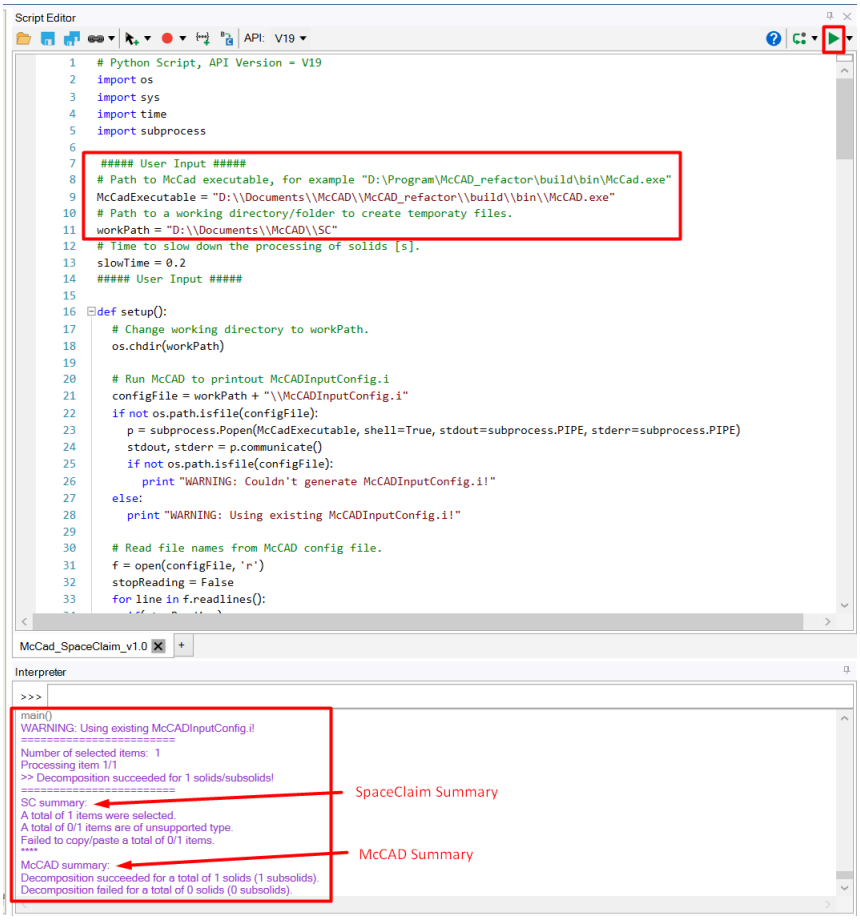



Figure 2: Example Input Solids

# 5 Example

This section will discuss how to run McCAD from a command line. The example input STEP file as well as the output files can be found in examples/collection_of_solids folder. The input file, SS_-793e-2.stp, contains a collection of random solids, as shown in figure (3); with steps, cylinders, tori, protrusions, depressions, etc. The solids from the file are assigned stainless steel material (SS) with a density of 7.93 g/cm$^3$, as indicated by the name of the STEP file.

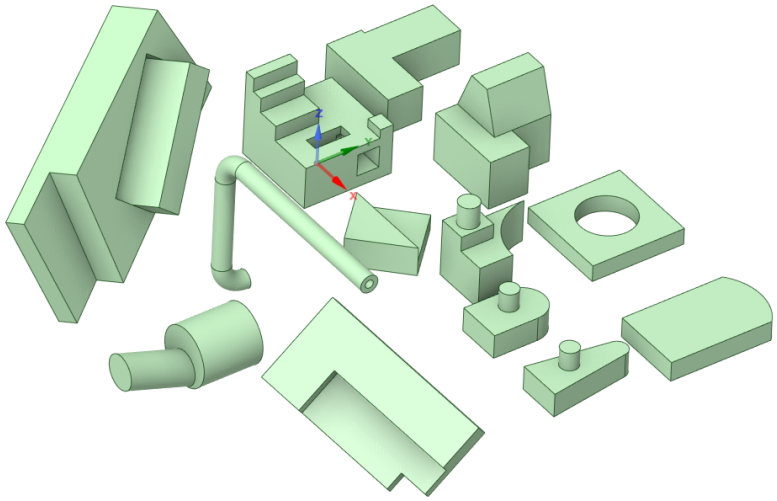


Figure 3: Example Input Solids

1. with the path to McCAD executable added to the PATH environment variable, the McCADInputConfig.i file could be generated by running McCAD, without any arguments, figure (4).

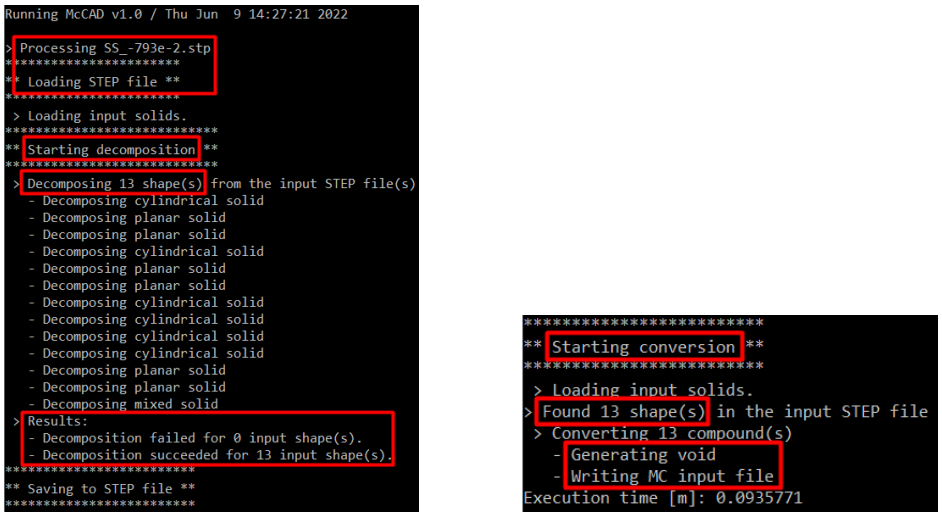
```
Running McCAD v1.0 / Thu Jun  9 14:25:42 2022

A template file, McCADInputConfig.i, with run parameters has been created in
D:\Documents\McCAD\test_McCAD\Benchmark
Execution time [m]: 0.000115512
```

Figure 4: McCAD Displayed Output

2. on McCADInputConfig.i file, the *inputFileName* should be changed to SS_-793e-2.stp. Also, *simplifyTori*, *convert*, and *voidGeneration* parameters should all be set to *true*.

3. McCAD is then run via the command "McCAD run". After successfully running McCAD there should be a total of five new files generated on disk. Figure (5) shows the displayed output of McCAD on screen for decomposition and conversion.




(a) (b)

Figure 5: McCAD Decomposition (a) and Conversion (b) Displayed Outputs

- the first file is produced by McCAD STEP file reading algorithm, SS_-793e-2Volumes.i, and contains a list of solid volumes and their names as read from the input STEP file, SS_-793e-2.stp.
- the second file is produced by McCAD decomposition algorithm, SS_-793e-2Decomposed.stp, and contains the decomposed input solids. Upon loading the file into any CAD software we could see the decomposed solids, as shown in figure (6).
- the third file is produced by McCAD conversion algorithm, MCFile.i. The file is the MCNP input file and contains cells, surfaces, and data cards in MCNP format, figure (7).
- the fourth file is produced by McCAD conversion algorithm, volumes.i. The file lists cell IDs, volumes of solids, and name on the CAD STEP file, figure (8).
- the fifth file is produced by McCAD conversion algorithm, voidCells.i. The file maps void to material cell IDs, figure (9).

4. Using MCNP plotter, figure (10) shows a plot of the MCFile.i and the corresponding CAD cross-section.

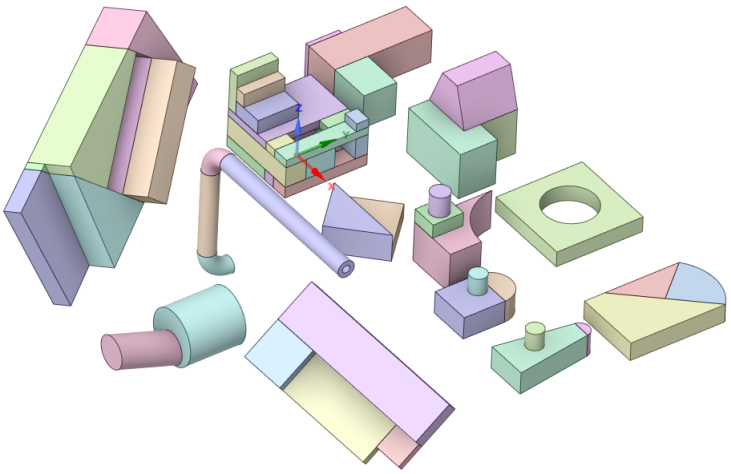


Figure 6: McCAD Decomposed Input Solids

```
McCAD v1.0 generated mcnp input file / Thu Jun  9 14:27:27 2022
C   * Material Cells      ----  54
C   * Material Surfaces   ----  128
C   * Void cells          ----  3
C   * Void Surfaces       ----  3
c ================================ Cell Cards ================================
c ============
c * Compound  : COMPOUND
c * Subsolids: 1
c * Material  : ss
c * Density   : -7.93
c * Cells     : 1 - 1
c ============
1     1  -7.93000 1 2 -3 -4 -5 6 7 Imp:N=1.0 Imp:P=1.0 Imp:E=0.0 TMP=2.53E-8
      $U=100000
c ============
c * Compound  : COMPOUND
c * Subsolids: 2
c * Material  : ss
c * Density   : -7.93
c * Cells     : 2 - 3
c ============
2     1  -7.93000 8 -9 10 11 -12 Imp:N=1.0 Imp:P=1.0 Imp:E=0.0 TMP=2.53E-8
      $U=100000
3     1  -7.93000 13 -14 10 -15 -11 Imp:N=1.0 Imp:P=1.0 Imp:E=0.0 TMP=2.53E-8
      $U=100000
c ============
c * Compound  : COMPOUND
c * Subsolids: 3
c * Material  : ss
c * Density   : -7.93
c * Cells     : 4 - 6
c ============
4     1  -7.93000 -16 -17 18 6 19 20 Imp:N=1.0 Imp:P=1.0 Imp:E=0.0 TMP=2.53E-8
      $U=100000
5     1  -7.93000 20 -19 -17 -21 6 Imp:N=1.0 Imp:P=1.0 Imp:E=0.0 TMP=2.53E-8
      $U=100000
6     1  -7.93000 -17 -19 6 21 -22 -19 21 Imp:N=1.0 Imp:P=1.0 Imp:E=0.0
      TMP=2.53E-8 $U=100000
c ============
```

Figure 7: McCAD Generated MCFile.i File

# 6 General Usage Notes

The following are general notes and best practices for using McCAD.

```
McCAD v1.0 generated volumes file / Thu Jun  9 14:27:27 2022
File contents: a map of cell IDs to CAD volume and name of the corresponding solid.
Column 1 is the cell ID, column 2 is the volume [cubic cm], and column 3 is the compound name.
1 1.20162E+00 COMPOUND
2 3.59000E-01 COMPOUND
3 6.55000E-01 COMPOUND
4 9.92320E-01 COMPOUND
5 3.20923E-01 COMPOUND
6 4.68008E-01 COMPOUND
7 6.54400E-01 COMPOUND
8 1.74406E+00 COMPOUND
9 6.54400E-02 COMPOUND
10 6.41431E-01 COMPOUND
11 1.73887E+00 COMPOUND
12 1.78503E+00 COMPOUND
13 1.20204E+00 COMPOUND
14 1.04096E+00 COMPOUND
15 7.04021E-01 COMPOUND
16 5.20248E-02 COMPOUND
17 3.41412E-02 COMPOUND
18 5.83682E-01 COMPOUND
19 1.83520E+00 COMPOUND
20 2.13120E-01 COMPOUND
21 4.47330E+00 COMPOUND
22 5.36000E-01 COMPOUND
23 5.20248E-02 COMPOUND
24 2.10487E-01 COMPOUND
25 1.52089E-01 COMPOUND
26 7.77544E-02 COMPOUND
27 1.21200E+00 COMPOUND
28 1.11720E+00 COMPOUND
29 1.88183E+00 COMPOUND
30 1.08619E+00 COMPOUND
31 3.54465E+00 COMPOUND
32 2.42556E+00 COMPOUND
33 8.14044E-03 COMPOUND
34 9.52734E-01 COMPOUND
35 2.40000E-02 COMPOUND
36 2.45000E-01 COMPOUND
37 8.22500E-02 COMPOUND
38 2.56500E-01 COMPOUND
39 4.86200E-01 COMPOUND
40 2.80500E-02 COMPOUND
41 3.92700E-02 COMPOUND
42 1.49600E-01 COMPOUND
```

Figure 8: McCAD Generated volumes.i File

```
McCAD v1.0 generated void-to-material cell IDs mapping file / Thu Jun  9 14:27:27 2022
File contents: a map of void cell IDs to the material cell IDs contained within.
Column 1 is the void cell ID and column(s) 2(+) is(are) the material cell ID(s).
55    2 3 7 8 9 12 13 14 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54
56    1 2 3 4 5 6 10 11 15 16 17 18 19 20 21 22 23 24 25 26 27 52
```

Figure 9: McCAD Generated voidCells.i File

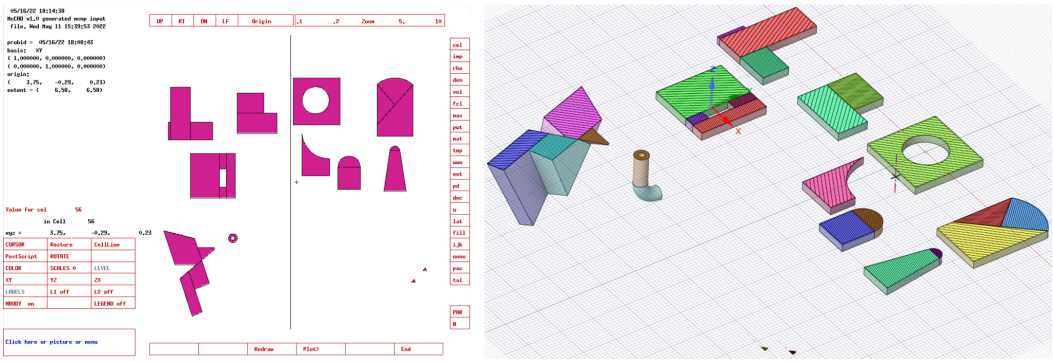


Figure 10: MCNP Plot of MCFile.i

- it is better to run decomposition and conversion separately!. This is the main reason why the conversion is disabled by default on the McCAD-InputConfig.i file. Checking the decomposed CAD solids for errors or interference is always advised!

- a good practice is to decompose the model few solids at a time. This ensures a smoother decomposition, as well as makes it easier to locate the troubling solid or part of the model.

- currently, complex hollow planar solids might result in large STEP files. It is a good practice to aid the decomposition by manually cutting the solid in two, three, or more subsolids then run McCAD on the subsolids.

- before running the conversion, it is better to check in CAD for any interferences in the model. This is especially needed if simplification of tori is enabled. A clean model will result in a conversion with less errors.

- before running McCAD conversion on the entire model, test the conversion on parts of the model separately. A good practice is to convert the solids that belong to a single material together. Test converting solids with another material. Then test converting both together. This way it is much easier to trace back the source of error and correct it in CAD rather than converting the entire model and try to figure out a solution to geometry errors.

- if the user wants a list of the CAD volumes, McCAD conversion STEP reader produces, in joint decomposition and conversion, mode, a volumes file with the format <file name>DecomposedVolumes.i. This file represents a list of volumes of the decomposed solids that will be converted. Running mcnp to calculate the stochastic cell volumes will produce volumes per cell. This way, the user will have a quantitative measure of the deviation of solid volumes due to decomposition and conversion; an important validation measure for converted CAD models.

- if McCAD decomposition and conversion are run separately, it is advised to rename the original volumes file so as not to get not overwritten by the volumes file produced by the conversion run. This way the user will have volumes file for the original solids as well as for the solids after decomposition.

# 7 Known Issues

Through extensive testing of McCAD decomposition and conversion core algorithms, some issues were found in relation to processing of solids. While some issues were fixed, not reported here, through debugging others persist and are in line to be addressed. Below is a list of the current known issues and how to manually fix them to produce a clean decomposed model to proceed with conversion. Keep in mind that you might not experience such issues in your model, but it is worth checking out.

## 7.1 Decomposition

- **Coordinate systems**:
  - *description*: the decomposition algorithm processes each of the input solids separately. This means that solids are decomposed according to their own local coordinate systems. When writing the output STEP file, the default coordinate system is used. As a result, if not all input solids have the same coordinate system that would result in some solids being displaced after decomposition.
  - *solution*: in SpaceClaim, one can check the local coordinate system by expanding the structure tree, selecting "Design > Move", then selecting solids, as shown in figure (11). Selected solids will display their system which can be compared to the global system of the model. If needed, by creating a new component and copy & paste the troubling solid, its new coordinate system will be aligned with the global one. Keep in mind that when creating components, the coordinate system is inherited from the parent, which means that sometimes one might need to replace a whole assembly to fix a coordinate system misalignment! Checking the whole structure tree is then advised.
- **Splitting of cylinders and tori**:
  - *description*: McCAD has the ability to split joined cylinders and tori, which often can be found in pipe work models. In some instances the code fails to detect the shared edge at the interface between the cylindrical and toroidal surfaces, leading to failure to split. In figure (12) an example is shown where a pipe was fully decomposed except at one of the sections, highlighted in orange, where the code failed to split the torus and cylinder.

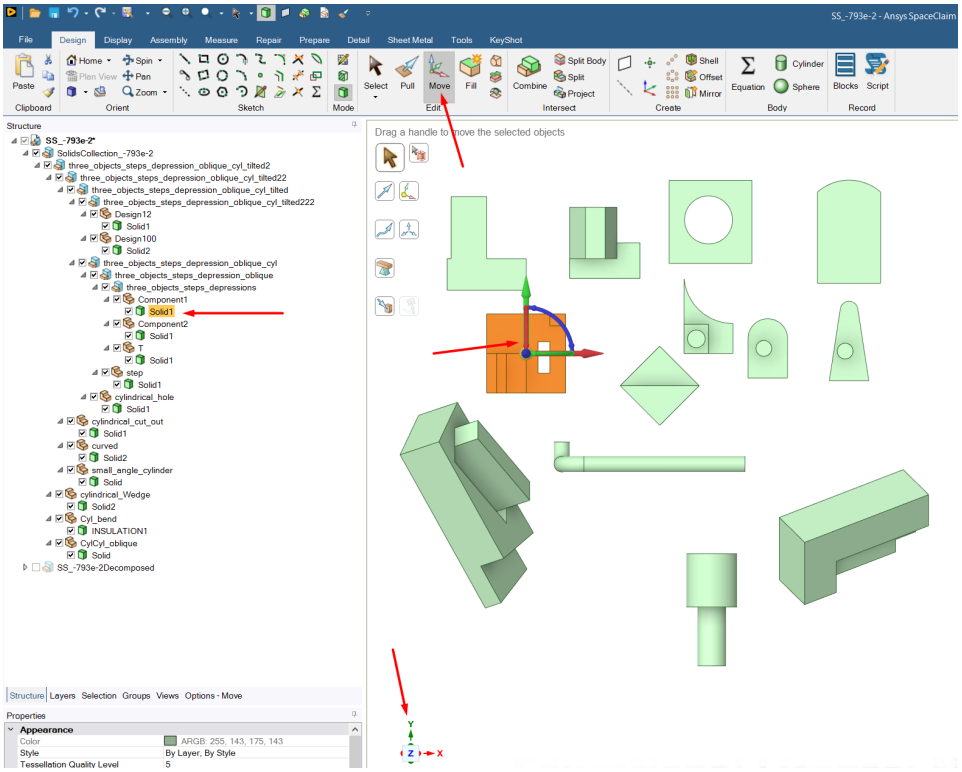
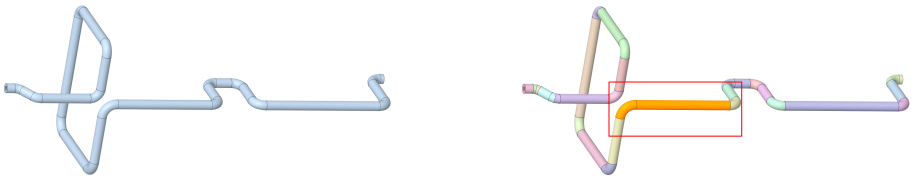



Figure 11: Coordinate System of a Solid in SpaceClaim

  - *solution*: the direct solution is to carefully examine the decomposed model and look for matching colors between sections where it was supposed to be split. Manually, in all CAD softwares, using cutting surfaces at the interface, the cylinder and torus can be split and saved along with the decomposed model.



(a) (b)

Figure 12: (a) Original Pipe in CAD and (b) McCAD Decomposed Pipe

- **Hollow solids**:
  - *description*: while not posing a problem for the decomposition algorithm, hollow solids pose a challenge to the run time and size

of the resulting STEP file(s). Figure (13) shows a pipe support structure. The structure is made primarily from planar surfaces and is hollow from the inside. Processing the solid as is in McCAD would take up some time and would result in a large STEP file. As a general rule: if McCAD took more than few minutes to decompose a planar surface then maybe a manual intervention on the user side is needed. This can be achieved by manually decomposing the solid into smaller pieces then run McCAD on those subsolids.

- *solution*: aide the decomposition by manually cut the hollow solid into three or four smaller sections which can then be processed in a reasonable time in McCAD.

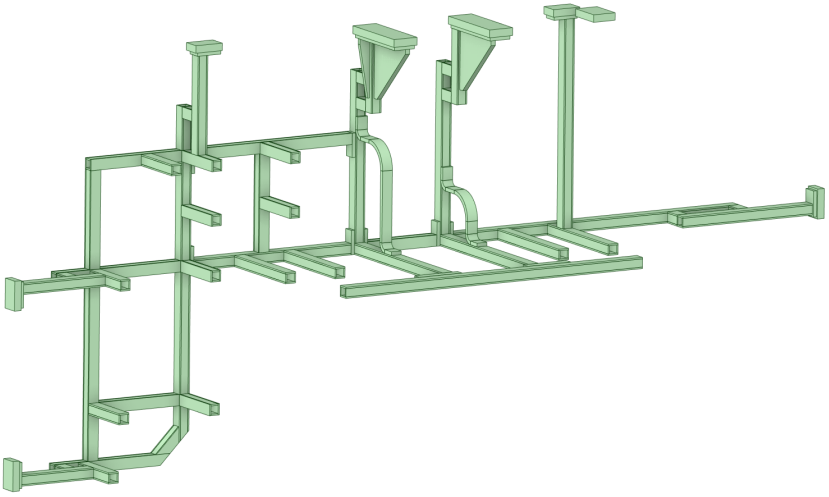


Figure 13: Coordinate System of a Solid in SpaceClaim

- **Solid names**:
  - *description*: when running decomposition solid names are retrieved from component names in the input STEP file. When saving the decomposed solids to STEP files, the names appear only on the assembly level while components take the name "COMPOUND". This is due to the difficulty to give names to components in the STEP writer in OCCT.
  - *solution*: loading the decomposed STEP files into a CAD software and manually changing the component names will alleviate such an issue and will result in actual component names appearing in cell cards in the generated MC file.

## 7.2 Conversion

None thus far!

# 8 Theory of McCAD Conversion

A comprehensive discussion on the inner workings of McCAD algorithms/classes for developers. Coming soon ...

## 8.1 Decomposition

## 8.2 Conversion