ORNL/TM-2022/2397

GEOMWRITER - A WORKFLOW FOR AUTOMATICALLY GENERATING MCNP GEOMETRY FOR NEUTRON GUIDE SIMULATIONS

Kyle B. Grammer

November 2022

OAK RIDGE National Laboratory

ORNL/TM-2022/2397

Neutron Technologies Division

A WORKFLOW FOR AUTOMATICALLY GENERATING MCNP GEOMETRY FOR NEUTRON GUIDE SIMULATIONS

K. Grammer

November 2022

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, TN 37831
managed by
UT-Battelle LLC
for the
US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

LIST OF FIGURES

ABBREVIATIONS

CAD	Computer-aided design
CSG	Constructive solid geometry
HFIR	High Flux Isotope Reactor
IGES	Initial Graphics Exchange Specification
OFF	Object file format
ORNL	Oak Ridge National Laboratory

ABSTRACT

This report describes the functionality of the `geomwriter.py` script, which is a python script for producing complex MCNP CSG geometries from the IGES CAD format. It is specifically written to handle neutron supermirror guides, but can handle more complex objects through utilizing a convex hull algorithm. This work flow reduces the amount of time spent on generating the surface definitions for a complex neutron guide system with hundreds of segments such that changes to the neutron guide system design in McStas can be rapidly incorporated into an MCNP model and integrated with the shielding geometry present in that model.

1 Introduction

The task of developing a large geometrical model of a complex system including shielding and neutron beam components for neutronics analysis of a neutron beamline is a time-intensive process and often an iterative process. The development of the optics for a neutron beamline in particular is a highly iterative process and there is a need to be able to make rapid changes to an MCNP [9] CSG model based on an updated McStas [10] instrument model.

In contrast to the McStas model which does not include any of the surrounding infrastructure, the MCNP geometry must completely and singly define the entire geometry. This means that the small gaps between successive guide segments must be determined as cells in the MCNP geometry, while McStas is free to ignore them. Additionally, the MCNP model should include the components of the real neutron guide system, which generally consists of an interior vacuum volume, a substrate and glass layer, a vaccum jacket, and a housing component and none of these components are present in the McStas model. As a result, a 100 rectangular segment guide system in McStas will contain 400 different cells and a total of 2400 surfaces when converted to CSG, though some of those surfaces can be eliminated as duplicates.

As the optics system might contain hundreds of separate guide segments, even a shift of the guide by a small translation or rotation results in the need to recompute the surface definitions for each of those 2400 surfaces. Since the CSG cannot contain any doubly defined regions, each of these 400 cells must be cut out, or subtracted, from shielding cells such that the geometry is completely and singly defined. A small translation or rotation of the guide system may also change which cells overlap, leading to the introduction of geometry errors and the requisite cuts for the updated model need to be checked and updated if necessary.

The goal of this set of scripts is to take a simple engineering representation of a guide geometry or shielding geometry and construct a working MCNP model from it quickly and to maintain the ability to quickly replace or update sections of the model. For this purpose, a new text based input language for reading CAD based IGES geometries that have been converted from McStas models as well as other simple geometric solids or OFF files. Once a working input file is constructed, small changes to the underlying engineering model are implemented in the MCNP geometry with generally only minor cleanup to fix any overlapping issues.

The primary python script, geomwriter, has been developed to process a new text based input language and outputs an MCNP input file. The text based input language gives the script instructions for how to process simple geometrical shapes, guide segments, tallies, transformations, materials, and CAD based IGES and OFF geometry files. The basic input line format is a series of option keywords followed by the values for those options. There are a number of identifiers that signify different line types with varying parameter sets involved for each type.

The input language can be divided into three types, position, geometrical, and auxiliary. Any input line that does not contain a "#" is treated as an input line, and lines that contain "#" in any position are ignored and treated as comments. Example input lines are shown in later sections as if the input spans multiple text lines, but this is simply for clarity in this document.

The position types signify the definition of MCNP transformations and reference positions for objects.

- `arm` (section 2.1)- specifying a location and an axis in space (the transformation of the **z** axis) that serves as an anchor point for geometrical bodies.
- `transform` (section 2.2) - specifying a `tr` card in MCNP, but the transformation can be determined by the script relative to a cell.

The geometrical types include instructions to read CAD files or produce geometrical objects and other cell-based output.

- `guidetaper` and `recttaper` (section 3.3) - specifying a tapered rectangular box with or without reflecting properties.
- `offile` (section 3.4) - specifying a geometrical body determined by an `.off` file that must be a "convex hull". More complicated bodies can be used by breaking them up into smaller convex bodies.
- `igeselement` (section 3.5) - specifying a geometrical body that is to be read from an `.igs` file.
- `sphere` (section 3.7) - specifies a spherical body in MCNP.
- `c(x,y,z)relative` (section 3.8) - specifying a cx, cy, or cz cylindrical object that is placed in the model with an MCNP `tr` card.
- `specialcase` (section 3.9) - specifying an arbitrary cell using existing surface numbers.

The auxiliary types generally include materials, transformations, instructions to read a file, variables, arms, and clean-up instructions.

- `read` (section 4.2) - a line instructing the script to read another input file line by line, allowing for easy addition or removal of components
- `import` (section 4.2) - a line instructing the script to read a subsection of another MCNP input file and incorporate it into the current model
- `material` (section 4.3) - specifying an `m` card in MCNP along with a density that is used for geometrical bodies specified later in the input.
- `thermal`, `sans`(section 4.3.1) - specifying whether a material has a thermal or SANS treatment applied to it.
- `reff` (section 4.4) - specification of supermirror reflective properties.
- `variable` (section 4.5) - specifies values that can be used to replace specific entries in multiple different elements.
- `tally` (section 4.6) - specifies different types of MCNP tallies.

Options on the input line are identified by a double dash that is immediately followed by the value of that option. Some types (arm and material) allow a specifically formatted list at the end of the input line that will be described below. In this document, input will refer to the input associated with the geomwriter script and output will refer to the output of the geomwriter script (which is an MCNP input).

1.1 Coordinate convention

The geometry and coordinate transformations use the convention for the axis of a neutron beam such that it is assumed that neutrons travel in the approximately $+\hat{z}$ direction from the source, that the floor is in the $-\hat{y}$ direction relative to the beam, and that $+\hat{x}$ is to the beam left direction. This assumption is preserved across all of the geoemtry definitions represented in this script. For example, when an arm with a new coordinate system is represented, the rotation matrix is determined by assuming the $+\hat{z}$ axis points along a new specified axis vector. Additionally, an arm is found for all guides and rectangular objects that first computes the upstream and downstream ends of the object along the $+\hat{z}$ direction, finding the vector that connects them, and using that as the axis for a transformation.

2 Positioning of components

The `transform` and `arm` types are the same except that a `transform` will be printed as a `tr` card in the resulting MCNP geometry file and an `arm` is used by the `geomwriter` script for the determination of surface definitions and the placement/alignment of cells. In both cases, there are two methods to determine the transformation of arm, named absolute and relative.

There are two cards common to absolute and relative transformations. Like in most instances, `idnum` specifies the ID number for this line. The `phase` card determines the rotation of the x-y plane about the z axis. It defaults to 0.0 if not specified.

2.1 Absolute transformation

The transformation takes an ID number to identify it and a phase, ϕ, that defaults to 0.0 determining a rotation in the $x - y$ plane. The parameters specifying the rest of the transformation are listed as 6 floating point numbers at the end of the type line, and these 6 numbers specify an origin in space, $O = \langle X, Y, Z\rangle$, followed by an alignment vector, $\mathbf{B} = \langle U, V, W\rangle$. The linear part of the transformation is given by the specified origin. The rotation matrix for the transformation is determined by finding the rotation matrix that takes the vector $\hat{z} = \langle 0, 0, 1\rangle$ and aligns it with the normalized alignment vector, $\mathbf{b} = \mathbf{B}/||\mathbf{B}||$. This is straight forward to do, but requires an initial assumption for the rotation of the $x - y$ plane that is determined by ϕ.

Let $\mathbf{v} = \hat{z} \times \mathbf{b}$ and let $c = \hat{z} \cdot \mathbf{b}$. The skew symmetric cross-product matrix of $\mathbf{v}$ is given by,

$$[v]_\times = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}. \tag{1}$$

The rotation matrix, $\mathbf{R}$, that aligns $\hat{z}$ with $\mathbf{b}$ is given by

$$\mathbf{R} = \mathbf{I} + [v]_\times + \frac{1}{1+c}[v]_\times^2, \tag{2}$$

where $\mathbf{I}$ is the identity matrix. This produces the correct matrix except in cases where $\hat{z} \cdot \mathbf{b} = \pm 1$, in which case $\mathbf{R} = \pm\mathbf{I}$.

The following line creates arm 500 at $O = \langle X, Y, Z\rangle$ pointing in the $\hat{z} = \langle 0, 0, 1\rangle$ direction. No output is generated for an `arm`.

```
--type arm --idnum 500 --phase 0.0 -- 0.2413 2.8702 295.092 0.0 0.0 1.0
```

The additional six values at the end of the line specify $O = \langle X, Y, Z\rangle$ and $\mathbf{B} = \langle U, V, W\rangle$. They must follow a double dash and a space and are interpreted as floating point numbers.

2.2 Relative transformation

A relative transformation takes an existing cell (sec. 3) number as input uses the transformation matrix associated with it. That cell must have already been processed by Geomwriter and it must also have a defined rotation matrix ($\mathcal{R}_{\text{rel}}$), linear transformation (O_{rel}), and an axis ($\mathcal{A}_{\text{rel}}$). The `IGES` and `recttaper` components both satisfy this condition, but objects like `.off` files do not. The transformation can be moved along the axis of the specified cell or translated in absolute coordinates with additional parameters.

The following line creates transform 101 relative to cell 9004 translated by 1 cm along the axis of that cell, and then translated by the tr_ cards. Cell 9004 in this case is a large cell placed at an anchor with an origin given by

$$O_{\text{cell}} = \langle 500.0, 443.32, 750.0 \rangle. \tag{3}$$

```
--type transform --idnum 101 --relative 9004 --phase 0.0
                 --cx 0.0 --cy 0.0 --cz 1.0
                 --trx -500.0 --try -443.32 --trz 0.0
```

This produces the following MCNP tr card:

```
tr101    0.000000 0.000000 751.000000
         1.000000 0.000000 0.000000
         0.000000 1.000000 0.000000
         0.000000 0.000000 1.000000
```

2.2.1 relative

The relative card specifies which existing geometrical cell this transformation will be relative to, and this will determine the rotation matrix, linear transformation, and the axis.

2.2.2 cx, cy, cz

These three cards specify the translation of the transformation along the axis of the relative cell from the origin of the relative cell, O_{rel}. Only the length of the vector specified by cx, cy, and cz is considered, where $\mathbf{C} = \langle cx, cy, cz \rangle$.

$$O_{\text{cell}} = O_{\text{rel}} + \mathcal{A}_{\text{rel}} \sqrt{\mathbf{C} \cdot \mathbf{C}} \tag{4}$$

2.2.3 trx, try, trz

The transformation can also be moved in absolute coordinates by supplying a vector in the trx, try, and trz cards. After the new origin is determined from the relative cell and the length of **C**, the new origin is determined using $\mathbf{T} = \langle trx, try, trz \rangle$

$$O_{\text{cell}} = O_{\text{rel}} + \mathcal{A}_{\text{rel}} \sqrt{\mathbf{C} \cdot \mathbf{C}} + \mathbf{T} \tag{5}$$

2.3 Applying transformations to surfaces

Since the number of surfaces for a complex MCNP geometry can be quite large, it is not possible to simply use a large number of transformation cards to position pieces of the geometry in the problem because MCNP is limited to 999 transformation cards. Geomwriter instead prefers to calculate the surface definition and print it without a transformation card and prefer to print transformations only when they are needed.

A generalized first or second order surface in MCNP can be represented with a gq surface as follows [9]

$$Ax^2 + By^2 + Cz^2 + Dxy + Eyz + Fzx + Gx + Hy + Jz + K = 0. \tag{6}$$

The process for transforming such a surface in MCNP [2] is implemented in Geomwriter in the following manner. First, the equation of the surface is represented by

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} M \begin{bmatrix} x \\ yz \\ 1 \end{bmatrix} = 0, \text{ where } M = \begin{bmatrix} A & \frac{D}{2} & \frac{F}{2} & \frac{G}{2} \\ \frac{D}{2} & B & \frac{E}{2} & \frac{H}{2} \\ \frac{F}{2} & \frac{E}{2} & C & \frac{J}{2} \\ \frac{G}{2} & \frac{H}{2} & \frac{J}{2} & K \end{bmatrix}. \tag{7}$$

For a rotation matrix, R, and translation, T, given by

$$R = \begin{bmatrix} r_{0,0} & r_{0,1} & r_{0,2} \\ r_{0,1} & r_{1,1} & r_{1,2} \\ r_{0,2} & r_{2,1} & r_{2,2} \end{bmatrix}, T = \begin{bmatrix} t_0 & t_1 & t_2 \end{bmatrix}, \tag{8}$$

two matrices can be constructed in order to determine the coefficients of the transformed surface

$$A_R = \begin{bmatrix} r_{0,0} & r_{0,1} & r_{0,2} & 0 \\ r_{0,1} & r_{1,1} & r_{1,2} & 0 \\ r_{0,2} & r_{2,1} & r_{2,2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \ A_T = \begin{bmatrix} 1 & 0 & 0 & t_0 \\ 0 & 1 & 0 & t_1 \\ 0 & 0 & 1 & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{9}$$

and the coefficients of the transformed surface are given by performing the following operation

$$M' = A_L^{\text{tr}} A_R^{\text{tr}} M A_R A_L. \tag{10}$$

When an arm or transform is applied to a surface in Geomwriter, it is applied to the matrix form of the equation 6 as described above and is returned as either a plane (if the second order terms are zero) or a gq surface otherwise.

3 Geometrical components

Geometrical components, or cells, are interpreted as MCNP CSG cells and there are several different types ranging from primitive boxes (`recttaper` and `guidetaper`) to complicated geometrical objects specified in `.off` or `.igs` files (`offile` and `igeselement`). All units are in centimeters, the usual unit for MCNP.

Once the surfaces of a complicated cell have been determined, Geomwriter must determine the sense of the cell relative to each surface. The centroid, **C**, of the cell is computed from the list of points that comprise the entire cell. For each surface, the difference between the centroid and a point on the surface, **S**, if found. The normal of the surface, $\hat{n}$, is then used to determine the sense of the surface relative to the center of the cell

$$\text{Sense} = \begin{cases} +, & \text{if } \hat{n} \cdot (\mathbf{C} - \mathbf{S}) \leq 0 \\ -, & \text{otherwise.} \end{cases} \tag{11}$$

Before describing the geometrical components in detail, there are a number of common cards associated with them that are introduced first.

3.1 Common cards

Geometrical components share a number of common cards that are interpreted in the same manner by each geometrical type.

3.1.1 ID Number

Like in most instances, `idnum` are used to specify the ID number for this line.

3.1.2 importance

The importance of the resulting cell is specified by the `imp` card, which takes an integer as a parameter.

3.1.3 idnumoffset

This parameter serves to move the surface numbers that will be associated with this cell to start at the specified value. The code checks whether the surface numbers are in use before using them, so using the same `idnumoffset` value for a block of associated geometrical objects will cause the entire block to use surfaces starting from the specified value.

3.1.4 cutfrom and elim/cut

Multiple `cutfrom` and `elim/cut` cards can be included on a line for a primary cell. The resulting primary cell with be “cut” from each cell number specified in `cutfrom` cards. Similarly, multiple `elim` or `cut` cards will “cut” other cells from the primary cell. Cells that will be cut from the primary cell should be specified before the primary cell in the input. Only the primary cell definition of a primary cell will be cut from another cell.

The following produces three cubes at anchor 1 with 5cm, 10cm, and 20cm dimensions. Cell 100 is a $5 \times 5 \times 5$cm cube with surfaces numbered from 101. Cell 200 is a $10 \times 10 \times 10$cm cube with surfaces numbered from 201, and it additionally will be `cutfrom` cell 300 and it will `cut` cell 100. Cell 300 is a $20 \times 20 \times 20$cm cube with surfaces numbered from 301 and with no additional cuts. An alternative specification would be to place `--cutfrom 200` on cell 100 and remove `--cut 100` from cell 200, but this illustration shows the `cut` and `cutform` cards in use together.

```
--type arm --idnum 1 --phase 0.0 -- 0.0 0.0 0.0 0.0 0.0 1.0
--type rect --idnum 100 --length 5.0 --height 5.0 --tapheight 5.0
            --width 5.0 --tapwidth 5.0 --idnumoffset 101
            --anchor 1
--type rect --idnum 200 --length 10.0 --height 10.0 --tapheight 10.0
            --width 10.0 --tapwidth 10.0 --idnumoffset 201
            --anchor 1 --cutfrom 300 --cut 100
--type rect --idnum 300 --length 20.0 --height 20.0 --tapheight 20.0
            --width 20.0 --tapwidth 20.0 --idnumoffset 301
            --anchor 1
```

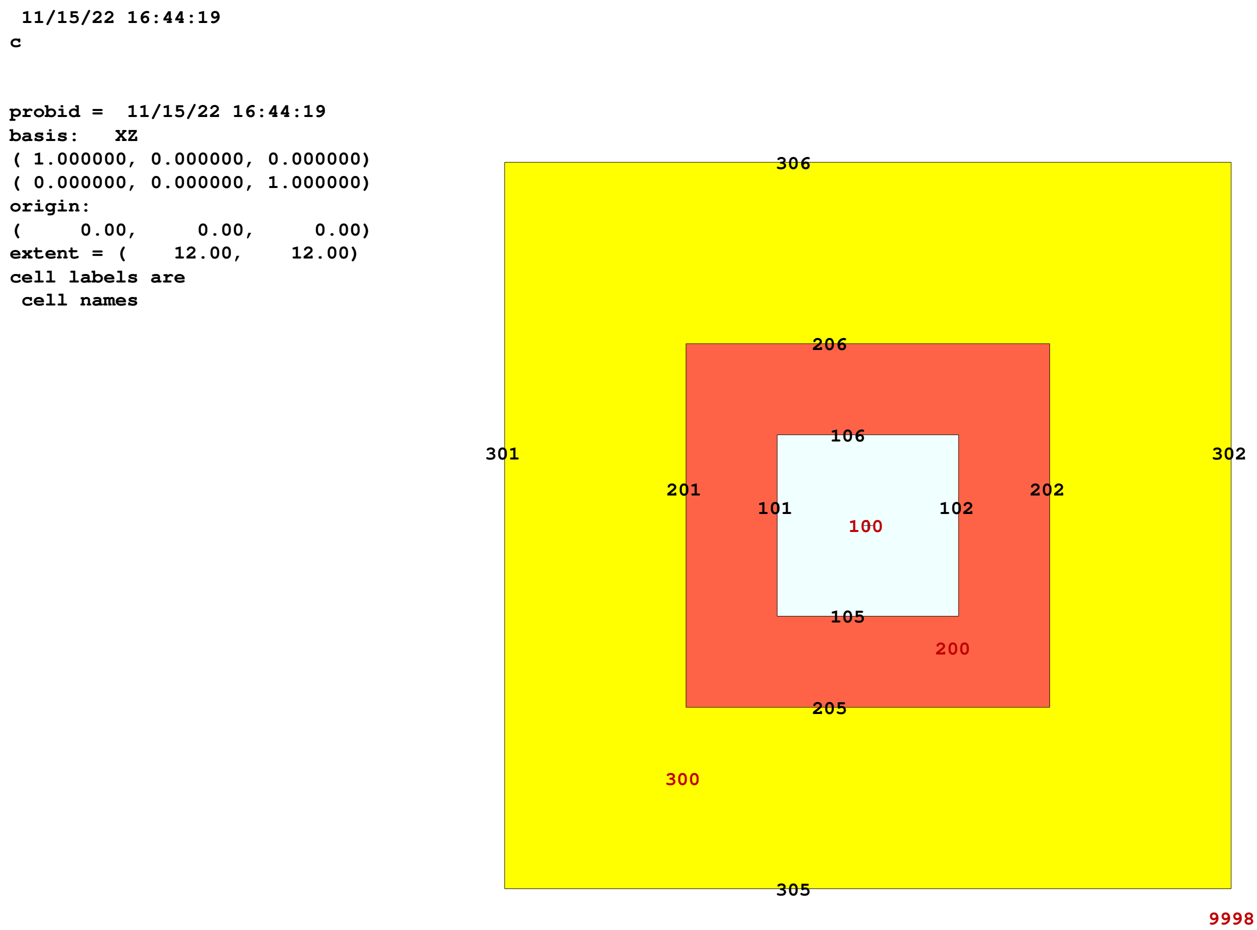

Figure 1. Illustration of nested box example showing the cut and cutfrom hierarchy. Cell 200 is orange and is cutfrom cell 300 (yellow) and cuts cell 100 (light blue). Cell numbers are in red and surface numbers are in black.

Figure 1 shows the illustration of this example in the $\hat{x} - \hat{z}$ plane, with the hierarchy that cell 100 is the inside, cell 200 is the middle, and cell 300 is the outside. The corresponding cell definitions are

```
  100     0     101 -102 103 -104 105 -106
  200     0     201 -202 203 -204 205 -206
              #( 101 -102 103 -104 105 -106 )
  300     0     301 -302 303 -304 305 -306
              #( 201 -202 203 -204 205 -206 )
```

For each of these, the primary cell definition is the 6 surfaces on the first line that define the cubes. Cell 200 was directed to `cut` cell 100 and the cell definition for cell 200 contains the complement of cell 100. Cell 200 was also directed to be `cutfrom` cell 300, and the cell definition for cell 300 thus contains the complement of the primary part of cell 200 but not also the complement of cell 100. In other words, complements are not propagated by the process of having one cell `cutfrom` another.

3.1.5 relative and anchor

The `relative` card value is a cell in the problem and the new resulting cell will be placed at a position relative to the cell specified here. An `anchor` card places the cell relative to an arm. These determine the cell rotation matrix, $\mathbf{R}_{\text{cell}}$, and translation, O, which provide the cell orientation and location, respectively.

3.1.6 comment

The comment card will be placed in the resulting MCNP input file as a comment to identify the cell. The value for this card should be a string inside quotation marks and may contain spaces.

3.1.7 material

The material card takes a material ID number as a parameter and uses that material number and the associated density for the cell. If no material card is specified, then vacuum is used.

3.2 Surfaces

Simple surfaces can be included by specifying the `surftype`, a surface number (`idnum`), an `anchor` designating the origin, and the parameter list. The `surftype` options are all of the standard MCNP surface mnemonic options excluding the torus and planes defined by points. The list of parameters is specified as a string with brackets and comma separation, which is a common way to list parameters for Geomwriter. Inside the routine for the surface definition, the rotation and translation `anchor` is applied using the routines discussed in section 2. The `params` option specifies the list of parameters that would appear on an MCNP surface definition card. The following lines are alternative ways to specify a cylinder with radius of 6 cm along the $\hat{z}$ axis that is then transformed to align with the unspecified `arm` 1. This allows the user to translate and rotate cylinders, cones, hyperboloids, etc. anywhere in the geometry by simply specifying the simplest form of the surface equation.

```
--type surface --surftype gq --idnum 100 --anchor 1
--params "[1.0, 1.0, 0, 0, 0, 0, 0, 0, 0, -36]"
--type surface --surftype c/z --idnum 100 --anchor 1
--params "[0, 0, 6]"
--type surface --surftype cz --idnum 100 --anchor 1
--params "[6]"
```

The base equation of this surface is given by

$$x^2 + y^2 - 36 = 0. \tag{12}$$

The first example uses the `params` option to define the coefficients of a `gq` in equation 6. The second defines a `c/z` cylinder parallel to the $\hat{z}$ axis with $(\bar{x}, \bar{y}) = (0, 0)$ and with a radius of 6. The third defines a `cz` cylinder (which assumes that it is on the $\hat{z}$ axis) with a radius of 6.

Consider another example with 3 different `arms`, where the first is aligned with the $\hat{z}$ axis, the second is aligned at a $\frac{\pi}{4}$ angle in the $\hat{x} - \hat{z}$ plane, and the third is aligned at $\frac{5\pi}{4}$ angle in the same plane. There are 4 different surfaces that define a cell. The first is a cone with an apex at $(0, 0, -5)$, the second is a paraboloid pointing downward with an apex at $(0, 0, 5)$, the third is a hyperboloid of one sheet centered at the origin, and the fourth is a plane at $z = -5$ to select the upper region of the cone. Consider also these same four surfaces aligned with the second and third arms above.

```
--type arm --idnum 1 --phase 0.0 -- 0.0 0.0 0.0 0.0 0.0 1.0
--type arm --idnum 2 --phase 0.0 -- -7.0 0.0 0.0 1.0 0.0 1.0
--type arm --idnum 3 --phase 0.0 -- 7.0 0.0 0.0 -1.0 0.0 -1.0
--type surface --surftype k/z --idnum 100 --anchor 1
       --params "[0, 0, -5, 1]"
--type surface --surftype gq --idnum 200 --anchor 1
       --params "[1, 1, 0, 0, 0, 0, 0, 0, 1, -5]"
--type surface --surftype gq --idnum 300 --anchor 1
       --params "[1, 1, -1, 0, 0, 0, 0, 0, 0, -1]"
--type surface --surftype pz --idnum 400 --anchor 1
       --params "[-5]"
```

The figure 2 illustrates the cell bounded by these surfaces aligned with the first `arm` in the center, the second arm on the left, and the third arm on the right. The figure shows an error in the black lines drawing the boundaries of each surface, but this is a plotting artifact in MCNP and the colors show the correct boundaries of the cells.

3.3 rect, guide, recttaper, and guidetaper

This is the first of the primary geometry definition cards, which inherits the common cards from the previous section. The `recttaper` card defines a rectangular prism with the capability of specifying different heights and widths at each end. The `rect` card is a simple rectangle with no taper and the `guide` versions are the same but include a reference to supermirror parameters.

3.3.1 transform

The ID number for an MCNP transformation will be applied to the surfaces of this cell.

3.3.2 length

Defines the length of the rectangular box along the $\hat{z}$ direction.

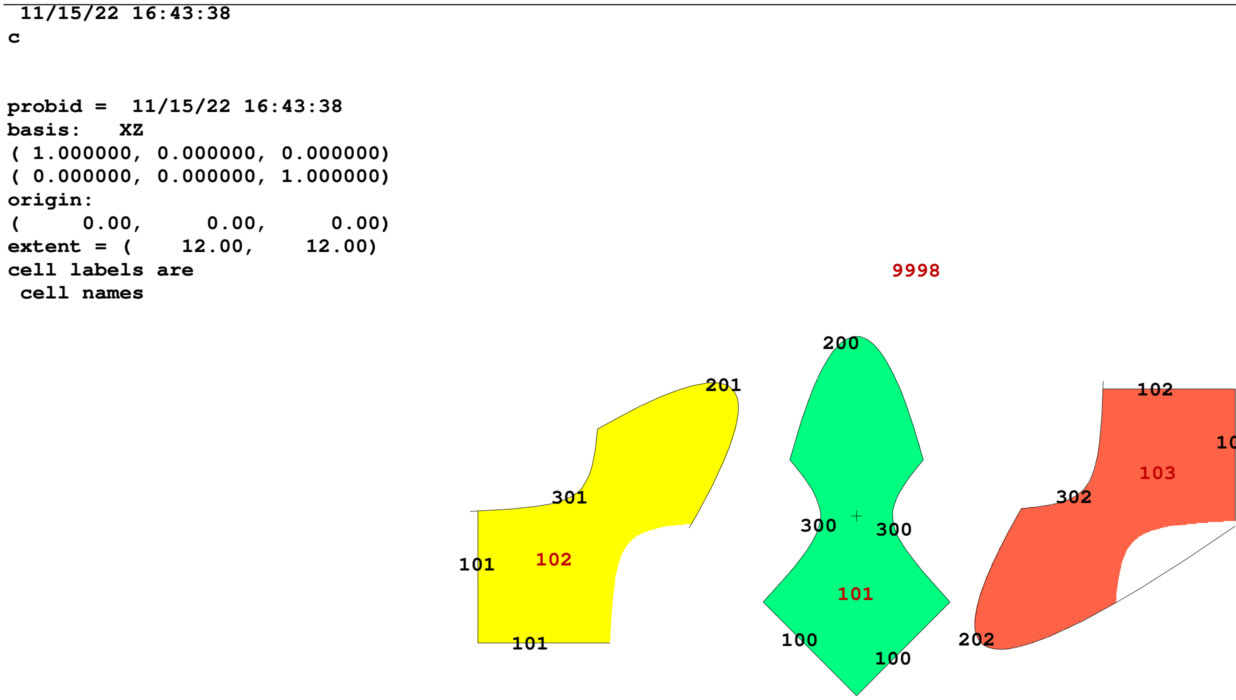

Figure 2. Surface example aligned with three different arms. Cell 101 (green) is aligned with arm 1, 102 (yellow) aligned with arm 2, and 103 (orange) aligned with arm 3.

3.3.3 height/tapheight and width/tapwidth

The `height` and `width` cards specify the height and width of the upstream end of the rectangular box. Likewise, `tapheight` and `tapwidth` specify the height and width of the downstream end of the rectangular box, but default to the same values as `height` and `width` for non-tapered components.

3.3.4 shift

An angular shift around the x, y, and z axes can be supplied to add an additional small rotation to the specifed `anchor` or `relative` alignment. The `shift` parameter can either be a list of three (or nine) floating point numbers inside quotation marks (`--shift "a b c"`) or a single floating point number (`--shift "b"`).If nine floating point numbers are supplied, they are treated as the elements of a rotational matrix. If three floating point numbers are supplied, they are handled as if they are all angles in radians, and will set up three rotation matrices, defined as follows (or only R_2 if a single angle is given)

$$\mathbf{R}_1 = \begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & \cos a & -\sin a \\ 0.0 & \sin a & \cos a \end{bmatrix} \tag{13}$$

$$\mathbf{R}_2 = \begin{bmatrix} \cos b & 0.0 & -\sin b \\ 0.0 & 1.0 & 0.0 \\ -\sin b & 0.0 & \cos b \end{bmatrix}, \tag{14}$$

$$\mathbf{R}_3 = \begin{bmatrix} \cos c & -\sin c & 0.0 \\ \sin c & \cos c & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}. \quad (15)$$

These are then applied to the cell rotation, $\mathbf{R}_{\text{cell,prev}}$, in the following order,

$$\mathbf{R}_{\text{cell}} = (\mathbf{R}_3(\mathbf{R}_2(\mathbf{R}_1\mathbf{R}_{\text{cell}}))). \quad (16)$$

3.3.5 nullrot

This nullifies a rotation axis from the cell rotation matrix, $\mathbf{R}_{\text{cell}}$ when determined only by `anchor` or `relative`. The parameter specified is a single character, "x", "y", or "z". This is meant to place an object relative to another cell, but aligned along a different axis. For instance, a shielding block should be aligned vertically with the $\hat{y}$ axis but might be placed relative to a guide segment that is tilted with respect to the $\hat{y}$ axis. Without `nullrot`, the shield block will also be tilted and resting precariously in space rather than standing vertically on the floor. The `nullrot` card allows the user to ignore an axis for the rotation matrix.

3.3.6 trx, try, trz

These cards move a cell along the axis in the same way that they move a transformation.

3.3.7 connector

The `connector` card is used to force a cell to attach to the cell that it is `relative` to, that is to say to make the junction between cells seamless and eliminate small gaps that might be present in a McStas file, for instance. A value of -1 on the connector card disables this functionality, otherwise all relative cells will be stitched seamlessly to the previous one.

3.3.8 Guide parameters

The guide refleciton ID can appear at the end of the line for the `guide` and `guidetaper` types. This should be the IDs of the `reff` cards containing the reflective properties for this guide segment. The appropriate reflection cards will be appended to the output and written later. It is possible for different surface properties to be represented here by specifying a list of 4 reflection IDs, and an example is shown later in this section. Unless exactly 4 are specified, the first reflection ID will apply to all surfaces for the segment. Generally, it is preferred to use IGES import to specify a guide than to use `guide`.

3.3.9 Manually changing surface numbers

The surface numbers for a cell can be manually changed by passing any of the following cards. The convention for naming them stems from the origins of the script being for neutron transport that is assumed to be directed in the (approxiamtely) $+\hat{z}$ direction.

- `ble` - The beam-left surface, which is normal to the $\hat{x}$ axis and has positive sense.
- `bde` - The beam-down surface, which is normal to the $\hat{y}$ axis and has positive sense.
- `use` - The upstream surface, which is normal to the $\hat{z}$ axis and has positive sense.
- `bre` - The beam-right surface, which is normal to the $\hat{x}$ axis and has negative sense.

- `bue` - The beam-up surface, which is normal to the $\hat{y}$ axis and has negative sense.
- `dse` - The downstream surface, which is normal to the $\hat{z}$ axis and has negative sense.
- `append` - Appends an additional surface to the cell description, including the specified sign.

3.3.10 Example input and output

First the 8 corners of the rectangular box are determined, with the first plane at $z = 0$ in the $x - y$ plane and the second plane at $z =$ length in the x-y plane. Once the translation and rotation matrix have been determined from the various cards that affect them, the points are rotated to the correct axial alignment and placed at the translation position. The equations for 6 MCNP `p`-type planes are then determined from the points, with a transformation number attached to them if the `transform` card was used. The script determines the geometrical center of the points in order to find the MCNP "sense" of each of the surfaces for the signs of the surfaces on the cell definition.

As one example of this type of geometrical construct, see this rectangle:

```
--type recttaper --idnum 9004 --length 250.0 --width 2000 --height 1100
            --tapheight 1100  --tapwidth 2000  --shift "0.0 0.0 0.0"
            --comment "enclosure" --idnumoffset 11001 --relative 9003
            --cx 0.0 --cy 0.0 --cz 250.0 --imp 1 --material 38
            --connector 1 --cutfrom 9998 --ble 10025 --bre 10029
```

which produces the following cell and surface definitions, including one of many cutout sections from cell 9004.

```
c enclosure
9004    38 -0.001290    +10025 -10029 +11027 -11028 +11029 -11030
                     #( -20474 +20475 +20476 -20477 +20478 -20479)
...
10025  p     1.000000 0.000000 0.000340 -99.996598
10029  p     0.998769 0.000000 -0.049594 99.381013
11027 py    -106.680000
11028 py    993.320000
11029 pz    750.000000
11030 pz    1000.000000
```

As an example of creating a guide in this manner, consider the following line with the accompanying `reff` card for a reflective function.

```
--type reff --idnum 1 0.99 0.0219 1.0 3.044 0.0025
--type reff --idnum 2 0.99 0.0219 2.0 3.044 0.0025
--type reff --idnum 3 0.99 0.0219 3.0 3.044 0.0025
--type reff --idnum 4 0.99 0.0219 4.0 3.044 0.0025
--type guidetaper --idnum 40016 --length 25.0 --width 3.9677 --height 4.9677
            --tapheight 4.957 --tapwidth 3.957 --idnumoffset 40000
            --shift "1.5710232196e-08 0.000614752086442 -1.08117911173e-06"
            --relative 40012 --cx 0.33519 --cy -0.044306 --cz 25.048
            --connector 1 --imp 10 --comment "NB4_Guide1_004" --cutfrom 40017 1 2 3 4
```

which produces the following guide cell, surfaces, refle cards, and reff cards.

```
c cccccccccccccccccc
c NB4_Guide1_004
40016     0    +40018 -40019 +40020 -40021 +40022 -40023
...
40018  p     0.999899 -0.000003 -0.014209 -3.506291
40019  p     0.999905 -0.000003 -0.013781 0.697989
40020  p     0.000000 1.000000 -0.000213 -3.438955
40021  p     0.000006 1.000000 0.000215 1.765325
40022  p     0.022600 0.001752 0.999743 552.722235
40023  p     0.023001 0.001756 0.999734 577.768005
...
refle17      40018 1 -40016
refle18      40019 2 -40016
refle19      40020 3 -40016
refle20      40021 4 -40016
...
reff1      0.990000 0.021900 1.000000 3.044000 0.002500
reff2      0.990000 0.021900 2.000000 3.044000 0.002500
reff3      0.990000 0.021900 3.000000 3.044000 0.002500
reff4      0.990000 0.021900 4.000000 3.044000 0.002500
```

3.4 Object File Format

The offile type reads from an .off file to create an arbitrary solid object. The card offile is the file name of the geometry for this card. The format describes a set of vertices and a set of polygons that describe the boundaries of the solid.

Lines that begin with # are treated as comments. The file begins with OFF. The first data line must be 3 numbers indicating the number of points (8 in the example), the number of faces (6 in the example), and the number of line segments (12 in the example, though this number is ignored). This is followed by a list of exactly 8 points, which will be used to determine the vertices of surfaces (numbered starting from zero). The surface definitions are last and include the number of vertices (4 in the example) followed by a list of the points defining the surface. Optionally, the .off file format supports a "color" at the end of each surface line, and the geometry script allows for 3 numbers to identify a set of surfaces as a "body" by giving them all the same "color". If the "color" information is not present, the file is treated as if it only has one "body". This allows for a single .off file to specify multiple MCNP cells in one file by identifying them as different bodies. When multiple bodies are in the file, they'll be numbered sequentially starting from the idnum card.

```
OFF
# box.off
# A box with 6 sides
8 6 12
1.0  0.0 1.0
0.0  1.0 1.0
```

```
-1.0  0.0 1.0
0.0 -1.0 1.0
1.0  0.0 0.0
0.0  1.0 0.0
-1.0  0.0 0.0
0.0 -1.0 0.0
#
4  0 1 2 3 1 0 0
4  7 4 0 3 1 0 0
4  4 5 1 0 1 0 0
4  5 6 2 1  1 0 0
4  3 2 6 7  1 0 0
4  6 5 4 7  1 0 0
```

The `.off` file should define a completely enclosed volume, but the script will attempt to close it by adding surfaces specified by the following two lines as the if only 4 surfaces are specified. However, this is not a universally correct solution and only works for certain `.off` files.

```
4  0 1 5 4
4  3 2 6 7
```

3.4.1 Positioning

An `.off` file positioned `relative` to a cell will have its alignment and origin determined in the same way as a relative transformation (section 2.2), and will utilize the `trx`, `try`, `trz` `cx`, `cy`, and `cz` cards in the same manner as relative transformations. This results in a rotation matrix and an origin that are determined before the file is read.

An `.off` file positioned at an `anchor` (an `arm`) will use the rotation matrix and origin specified by that arm. The `cx`, `cy`, and `cz` cards will be ignored, so the positioning along the `arm` should be handled when creating the arm.

When an `.off` file is initially read, the points will be placed at the appropriate `arm`, be that `relative` to another cell or at an `anchor` to an `arm`. Thus a single `.off` file can be replicated in the geometry simply by re-reading it and placing it at a different anchor.

3.4.2 scalefactor

The `scalefactor` card changes the units of the points in the off file, otherwise it will be assumed to be in MCNP units (centimeters). This is applied at the reading stage as a multiplicative factor.

3.4.3 skipredundancy

If the `skipredundancy` is provided, the script will bypass the check that a generated surface is redundant. This means that redundant surfaces may exist in the resulting geometry file, but surface number patterns that might be counted on for hard-coding cell modifications will be preserved.

3.4.4 guide

An `.off` file can be used as a rectangular guide. The first 4 surfaces of the guide should have a list of arguments at the end of the line to flag the appropriate `reff` card.

3.4.5 Convex hulls

Instead of specifying `offile` as the type, `offilehull` can be specified. In this case, the point cloud will be processed through a convex hull algorithm that produces the unique convex hull that surrounds the point cloud. This allows for the approximate incorporation of a complicated object without requiring the object to be sliced into smaller convex sections.

3.4.6 Slicing along an axis

Instead of specifying `offile` as the type, `offileslice` can be specified and a `binbounds` list can be passed as an option.

```
--binbounds "[y, 10, 200, 500, 1000, 1500, 2000]"
```

In this case, the point cloud will be sliced along the specified axis (first entry) at the values specified (the rest of the list). The intersection of all facets of the `.off` file with each specified plane is found. If the intersection is along a line segment of the surface, then that point is saved to a sub-list of points that will comprise individual cells of the sliced geometry.

3.4.7 Manually changing surface numbers

Like for rectangular geometrical features, the surface numbers can be altered. However, there are only two options.

- `append` - Appends an additional surface to the cell description, including the specified sign.
- `remove` - Removes all instances (positive or negative sense) of a surface from the cell description.

3.4.8 Example input and output

The following is a line that imports an `.off` file and cuts it from an unspecified different cell (10003). It contains an unspecified anchor (10000) and is comprised of an unspecified material (30) and has an importance of 3. The corresponding `.off` file is shown below.

```
--type offile --offile 7970bulkheadsA.off --idnum 10004 --idnumoffset 10000
         --material 30 --cx 0.0 --cy 0.0 --cz 0.0 --imp 3
         --scalefactor 1.0 --anchor 10000 --cutfrom 10003
         --comment "Bulkheads A_USBLCorner"
```

and which leads to the following output (some surfaces have been omitted):

```
c cccccccccccccccccc
c Building D
10003    38 -0.001290    +11004 +10047 -10048 -10036 +10049
         -10037
```

```
         #( +11004 +10000 -10001 -10002 +10003 -10004)
c cccccccccccccccccc
c Bulkheads A_USBLCorner
10004    30 -3.930000    +11004 +10000 -10001 -10002 +10003 -10004
...
10000  p     0.429286 0.000000 0.903168 1454.684522
10001  p     0.999151 0.000000 0.041187 -32.169013
10002  p     0.404988 0.000000 0.914322 1542.618778
10003  p     1.000000 0.000000 0.000312 -157.007215
10004 py    112.320000
11004 py    -106.680000
```

The .off file used to specify the cell above is shown here. Line 1 contains the keyword OFF. Lines beginning with a "#" are comments. Line 4 contains the number of vertices (8), the number of faces (6), and the number of edges (0 because this is an ignored parameter). The next set of lines is the list of vertices in $\langle XYZ \rangle$ format and which are referred to by number starting at 0. There is then a blank line followed by the list of faces in the format of the number of vertices defining that face followed by a list of the vertices that are on that face. In this example, each face is over-determined as having 4 vertices when 3 would be sufficient. The script will use only the first vertices in the list to define the face.

```
OFF
# 7970bulkheadsA.off
# The 7970 guide hall with extension (should be placed relative to source arm in input)
8 6 0
-157.5328937919 -106.68 1685.52294143183
-100.559903081318 -106.68 1658.44303302053
-103.636442542127 -106.68 1733.0773065487
-157.55517315637 -106.68 1756.95999737673
-157.5328937919 893.32 1685.52294143183
-100.559903081318 893.32 1658.44303302053
-103.636442542127 893.32 1733.0773065487
-157.55517315637 893.32 1756.95999737673
#
4 0 1 2 3
4 0 1 4 5
4 1 2 5 6
4 2 3 6 7
4 3 0 7 4
4 4  5  6 7
```

To illustrate the difference between different representations of an .off file, an example of a sword is used from Holmes3d [6]. The CAD representation is shown in figure 3, which shows significant detail along the length of the sword. The convex hull representation is shown in figure 4, and which removes most of the detail since the convex hull is the bounding surface of the point cloud. One slicing representation is shown in figure 5 where the sword has been sliced at 6 positions along the length, and which preserves some of the structure of the hilt and cross guard.

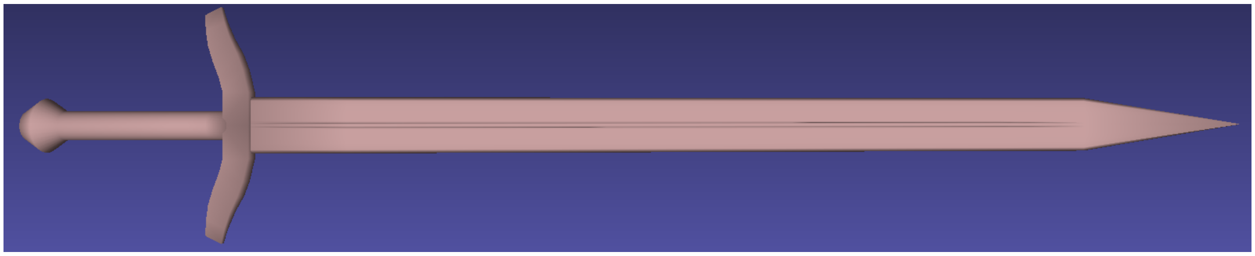

Figure 3. Example of a sword as displayed in CAD software.

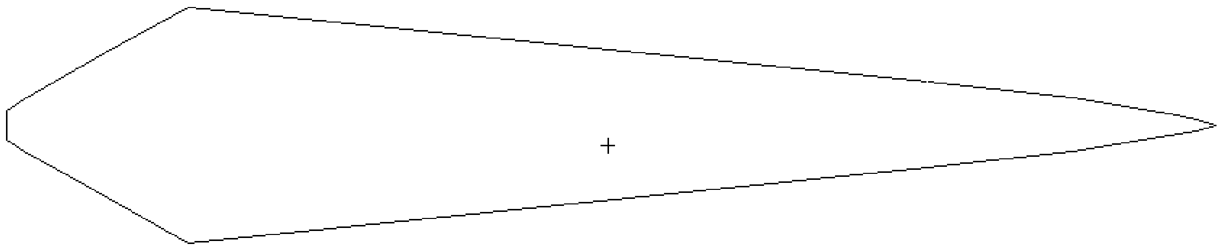

Figure 4. Example of a sword as interpreted by Geomwriter as a convex hull.

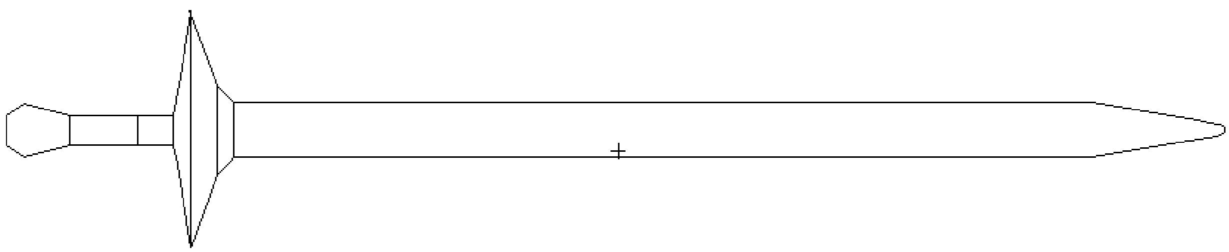

Figure 5. Example of a sword as interpreted by Geomwriter with slices along the length of the sword.

3.5 IGES components

An IGES component can consist of an isolated `.igs` file or a zip archive of `.igs` files, and either form can be imported. An IGES component is specified with the `igeselement` card. The handling of an IGES zip repository manually is quite cumbersome and best handled by a separate pre-processor script that produces an input for geomwriter. It's in this pre-processing script that facility specific assumptions can be made. Examples of facility dependent assumptions include:

- producing glass, vacuum jacket, and casing cells
- stitching together gaps between adjacent guide segments
- processing segmented or other specially shaped guide segments
- cutting the casing cells out of surrounding shielding cells

Generic IGES compatible files are not necessarily able to be imported because only a small subset are compatible with geomwriter.

3.5.1 File format and geometry generation mechanism

The `.igs` file should contain a series of line segments (type 110, LINE) that connect points and which are used to define the vertices of planes, plane equations (type 108, PLANE) which are not used in the script because the equations for planes are determined separately, boundaries (type 141, BOUNDARY) which list the LINE type and PLANE type entires that define a boundary, and boundary surfaces (type 143, BND_SRF) which further define the surface as a boundary of the volume. Additionally, the script reads some null lines (type 0) of certain formats in order to define the reflective properties of a guide and to define the end caps of a complete cell. The end caps are written identically to a boundary surface but have type 0 so that they will be ignored by CAD software. The reflective properties of a guide are defined on a null line (NEUT_SM data type) with a series of floating point numbers in order of R_0, q_c, m, α, W. Boundary surfaces that have reflective properties include a pointer to the appropriate NEUT_SM line. The file also contains a header with information that includes the name of the object and the units that are being used (14th entry in the global section).

Since MCNP geometry is defined by surfaces rather than vertices, the vertices may not necessarily lie on the boundaries of the object. The vertices determine the equation of the plane in a consistent way in order to properly define the sense of the cell relative to surfaces for the MCNP geometry, which means that the plane equation given on type 141 lines is ignored and the equation of the plane is calculated by the script from the list of vertices found on type 110 lines. When a BND_SRF line points to a NEUT_SM line [3], the script checks whether that exact reflective function already exists, creates one if it does not exist, and adds the appropriate `reff` and `refle` lines for the output of the script.

3.5.2 igeszip

This is the filename of a `.zip` repository containing IGES files. The zip file can be in sub-directories below the working directory.

3.5.3 igessubfile

This is the file name in the `.zip` repository or extracted in a directory that contains the information for a specific element. If the `.zip` repository contains a subfolder, the complete path of the file in the repository should be specified here.

3.5.4 stddev

MCNP will optimize away redundant surface definitions. In general, surfaces are specified in MCNP according to the equation of a plane,

$$Ax + By + Cz - D = 0. \tag{17}$$

This is a problem in the case of neutron guides, because the reflective properties are not included in this optimization, and the input will fail to run. In order to avoid this, the `stddev` card can be supplied as a floating point number that defines the σ for a Gaussian random distribution, and each reflective surface in the IGES file will have a Gaussian random number added to the D parameter for the surface in order to make them unique.

3.5.5 expand

The `expand` card will expand each BND_SRF of the object by the specified value in the direction normal to the surface such that the center axis of the object remains unchanged. The end caps will not be expanded. This provides a clean mechanism for using the same `.igs` file to define an inner volume and a series of shells around it or to allow for small modifications to the dimensions of the geometry in case MCNP optimizes away redundant guide surfaces if the user prefers not to use the `stddev` option.

3.5.6 mirror

The reflective properties will be included if a value of 1 is passed on the `mirror` card and will be ignored if a value of 0 is passed. This facilitates the use of a guide `.igs` file for the internal guide vacuum volume as well as concentric shells that exist around without flagging the shell as reflective. If a value of 1 is given and the `.igs` file does not contain any supermirror information, default values of $R_0 = 0.990$, $Q_c = 0.022$, $m = 8$, $\alpha = 3.044$, and $W = 0.003$ will be used.

3.5.7 Manually changing surface numbers

The surface numbers for a cell can be manually changed by passing any the same set of cards as those found in section 3.3.9. There is an additional option available here:

- `finalsurface` - manually replaces the last surface (6th) in the cell definition with a different surface.

This functionality should be avoided for non-rectangular guide segments.

3.5.8 igessegment

The list of boundaries to use from the IGES file is specified as a comma separated list inside square brackets, ie. ”`--igessegment "[20, 30, 40, 50]"`”. When specified, only the list of boundaries given will be used. This allows for the isolation of smaller bodies within a single IGES file, such as a segmented approximation of a true curved surface, and is also used to produce channeled guides. All of the boundaries in the IGES file are used to create the cell when this parameter is not specified.

3.5.9 File format and processing

Simple rectangular volumes are described by a series of line segments (IGES 110) that bound boundaries (IGES 141) described by planes (IGES 108) and boundary surfaces (IGES 143). The specification of neutron supermirrors was implemented [3] via null entities (IGES 0) such that the supermirror parameters are ignored by CAD software but can be read by this script. The parameters on a supermirror line are given in order of R_0, q_c, m, α, W, where R_0 is the low-q reflectivity, q_c is the critical angle, m is the supermirror parameter, α is the slope above q_c, and W is the cutoff width, and the appropriate supermirror parameters are linked to a specific guide surface boundary via a boundary surface line (IGES type 143) [3]. In the case that a volume is stated to be a guide (indicated by the mirror parameter) and there is no null entity line with supermirror parameters, the default values indicated previously will be used.

Each guide `.igs` file is processed to find all points, planes, and auxiliary information for each plane (supermirror properties) and returned as a single dictionary for each component. The end cap surfaces are used directly if they are present in the `.igs` file, otherwise they are determined with an algorithm. The centroid of the two endcap surfaces is subtracted to determine the axis of the guide segment, and the centroid of the upstream endcap is used as the reference translation for the guide segment. Together, these vectors are used to determine the appropriate transformation that would place an object at the beginning of the guide segment and facilitate the placement of objects relative to a guide segment by number.

In some cases, an IGES system is determined to be too complex to handle cleanly. In this case, the points comprising vertices are entered into the SciPy ConvexHull algorithm to produce a simple convex hull from the points. The simplices of the hull represent the facets of the surface, each which becomes a plane in the MCNP geometry.

The sense of an MCNP cell definition relative to the surfaces comprising the boundaries in the IGES file is determined by finding the vector between the centroid of the body and the centroid of each facet.

3.5.10 Example input and output

The following lines read a particular `.igs` file from a specified `.zip` repository. The file contains a guide segment, and cell 20000 will be a guide while cell 20001 will be the glass volume concentric with the guide in cell 20000 and expanded by 0.1 cm. The complete IGES file referenced here is shown in Appendix A.

```
--type igeselement --igeszip "NB3_BioSANS_IGES_20201007.zip"
           --igessubfile "NB3_BioSANS_IGES_20201007/NB3_guide0179_20201007.igs"
           --comment "NB3_Guide1_1 20200727.101012" --idnum 20000
           --idnumoffset 20000 --imp 10 --cutfrom 20001
           --material 0 --stddev 0.001 --mirror 1
--type igeselement --comment "NB3_Guide1_1 20200727.101012 glass"
           --igeszip "NB3_BioSANS_IGES_20201007.zip"
           --igessubfile "NB3_BioSANS_IGES_20201007/NB3_guide0179_20201007.igs"
           --idnum 20001 --idnumoffset 20001 --imp 10 --cutfrom 20002
           --material 13 --expand 0.1 --mirror 0
```

```
c cccccccccccccccccc
c NB3_Guide1_1 20200727.101012
20000     0    -20000 +20001 +20002 -20003 +20004 -20005
c cccccccccccccccccc
c NB3_Guide1_1 20200727.101012 glass
20001    13 -2.200000    -20006 +20007 +20008 -20009 +20010 -20011
           #( -20000 +20001 +20002 -20003 +20004 -20005)
...
20000  p     0.000000 0.999962 -0.008727 1.896115
20001  p     0.000000 0.999962 -0.008727 -2.103889
20002  p     0.999928 -0.000105 -0.012036 -2.377187
20003  p     0.999939 -0.000096 -0.011036 2.102169
20004  p     0.011537 0.008726 0.999895 477.346449
20005  p     0.011537 0.008726 0.999895 502.346440
20006  p     0.000000 0.999962 -0.008727 1.996160
20007  p     0.000000 0.999962 -0.008727 -2.203841
20008  p     0.999928 -0.000105 -0.012036 -2.476130
20009  p     0.999939 -0.000096 -0.011036 2.201213
```

```
20010  p     0.011537 0.008726 0.999895 477.346449
20011  p     0.011537 0.008726 0.999895 502.346440
...
refle5      20000 2 -20000
refle6      20001 2 -20000
refle7      20002 2 -20000
refle8      20003 2 -20000
reff2      0.990000 0.022000 8.000000 3.044000 0.003000
```

3.6 Bandwidth choppers

The `chopper` card creates a cylindrical wheel with an angled cutout that will rotate at a defined frequency. Use of this feature requires a modified MCNP executable [4]. Geomwriter requires the user to specify the `rota` card separately and will create the appropriate `dchp` card. A chopper requires a transformation or a relative cell to place it in the correct location in space. The user must also create a transformation that will hold the angular position of the chopper wheel as well as a `rota` card for the for the rotational parameters. The rotational parameters must be specified in order for the script to process the input, but the use of the chopper extension can be disabled by manually commenting out the `rota` and `dchp` cards in the MCNP input.

3.6.1 radius

The `radius` card is the outer radius of the wheel in centimeters.

3.6.2 height

The `height` card is the inner radius of the wheel in centimeters.

3.6.3 transform

The `transform` card is the ID of an MCNP transformation at which the chopper will be placed.

3.6.4 relative

The `relative` card is the cell ID of a cell or arm that the chopper transformation will be placed relative to.

3.6.5 angle

This is the opening angle of the chopper wheel in degrees.

3.6.6 Trailing arguments

The trailing arguments at the end of the line define the `dchp` card output. The order of these parameters is `dchp` key ID, cell number, rotational position transformation number, step size in time, and a flag. The script will create a `dchp` card with key ID and key ID+1 for each cell (the wheel and the cutout).

3.6.7 Example input and output

The chopper card creates two cells. The first is the cutout section and the second is the chopper wheel with the cutout section removed. Additionally, the appropriate `dchp` cards are generated for each cell, and the required transformations and `rota` card are generated with the appropriate cards.

```
--type trans --idnum 353 --phase 0.0 -- 296.544 0.0 3523.87
                        -0.137084176072 0.0 0.990559401889
--type chopper --idnum 27224 --idnumoffset 27224 --cx 0.0 --cy 0.0 --cz 0.0
          --radius 25.002 --length 1.0 --height 15.002 --cutfrom 25725
          --imp 10 --material 49 --transform 353
          --angle 18 27225 27224 354 1e-6 3
--type trans --idnum 354 0.0 0.0 0.0 0.0 0.0 1.0
--type rota --idnum 27224 60.0 0 354
```

The following output is generated for the above input.

```
27224     0      -27224 +27225 -27226 +27227 +27228 -27229
27225    49 -7.890000    -27226 +27228 -27229
              #( -27224 +27225 -27226 +27227 +27228 -27229)
...
27224 353 p      0.987688 -0.156434 0.000000 0.000000
27225 353 p      0.987688 0.156434 0.000000 0.000000
27226 353 cz      25.002000
27227 353 cz      15.002000
27228 353 pz      -0.500000
27229 353 pz      0.500000
...
tr353    296.544000 0.000000 3523.870000
         0.990559 0.000000 -0.137084
         0.000000 1.000000 0.000000
         0.137084 0.000000 0.990559
tr354    0.000000 0.000000 0.000000
         1.000000 0.000000 0.000000
         0.000000 1.000000 0.000000
         0.000000 0.000000 1.000000
...
dchp27225      27224 27224 354 0.000001 3
dchp27226      27225 27224 354 0.000001 3
rota27224      60.000000 0.000000 354
```

3.7 Spherical components

Simple spherical objects can be created with the `sphere` card, which will result in the creation of two cells defining the inside and outside of the sphere and is useful for defining the outside world.

3.7.1 radius

The radius of the sphere is given in centimeters on the `radius` card.

3.7.2 transform

The `transform` card places the center of the sphere at an MCNP transform card.

3.7.3 Example input and output

The following input line creates a 350 m sphere,

```
--type sphere --idnum 9999 --idnumoffset 9999 --radius 35000 --imp 0
```

and two cells will be created from this, one for the inside and one for the outside. The inside region can then be used to cut other cells from the rest of the problem, and one such example is included here:

```
9998 0 -9999
       #( +11001 -11002 +11003 -11004 +11005 -11006)
9999 0 +9999
...
9999   so     35000.000000
```

3.8 Cylindrical components

There are three different type cards that will trigger a cylindrical geometrical object. They are `cx`, `cy`, and `cz`, which create a cylinder aligned with the $\hat{x}$, $\hat{y}$, or $\hat{z}$ axes, respectively. The initial definition of the cylinder will be defined such that the origin is at the center of the cylinder and a transformation is used to place it in space in the geometry.

There is a second way to specify a cylinder through the types `cxrelative`, `cyrelative`, and `czrelative`. These will take either an `anchor` or a `relative` cell and place the cylinder relative to them.

3.8.1 radius

The radius of the cylinder is given in centimeters on the `radius` card.

3.8.2 length

The length of the cylinder is given in centimeters on the `length` card.

3.8.3 transform

The `transform` card places the center of the cylinder at an MCNP transform card.

3.8.4 trx, try, trz

These cards move the cylinder along its axis by the indicated distance in centimeters. Only the card matching the cylinder type (`trx` for `cx`) is used and the other two are ignored.

3.8.5 relative

The cell will be placed relative to the specified cell according to the procedure in section 2.2, up to and including the use of the cx, cy, and cz and trx, try, and trz cards.

3.8.6 anchor

If an anchor is specified, then it must appear on an arm card and the cylinder will be placed at that location.

3.8.7 Manually changing surface numbers

The surface numbers for a cell can be manually changed by passing any of the following cards. This allows for the end-cap surfaces of the cylinder to be modified.

- use - The plane to which the cylinder has positive sense.
- dse - The plane to which the cylinder has negative sense.

3.8.8 Example input and output

The following is an example cylinder parallel to the $\hat{y}$ axis located at a specific transformation.

```
--type arm --idnum 901 -- -0.166375 3.4924375 445.230525 0.0 0.0 1.0
--type cyrelative --idnum 974 --mat 6 --rad 30.5 --idnumoffset 970 --transform 901
                  --length 100 --imp 1 --cutfrom 9002 --anchor 901
                  --cx 0.0 --cy 0.0 --cz 0.0
                  --trx 0.0 --try -50.0 --trz 0.0
                  --comment "Main shutter"
```

```
c cccccccccccccccccc
c Main shutter
974     6 -3.110000    -1012 +1013 -1014
...
1012 901 c/y     0.000000 0.000000 30.500000
1013 901 py     -50.000000
1014 901 py     50.000000
...
tr901    -0.166375 3.492437 445.230525
          1.000000 0.000000 0.000000
          0.000000 1.000000 0.000000
          0.000000 0.000000 1.000000
```

3.9 Manually defined cells

A 6 sided object can be defined by using the specialcase type. The surface cards are used to supply the appropriate surface numbers.

```
--type specialcase --idnum 978 --material 0 --imp 10 --cutfrom 974
                   --use 1005 --dse -1012 --ble 1000 --bre -1001
                   --bde 1002 --bue -1003
```

- `ble` - The beam-left surface, which is normal to the $\hat{x}$ axis and has positive sense.
- `bde` - The beam-down surface, which is normal to the $\hat{y}$ axis and has positive sense.
- `use` - The upstream surface, which is normal to the $\hat{z}$ axis and has positive sense.
- `bre` - The beam-right surface, which is normal to the $\hat{x}$ axis and has negative sense.
- `bue` - The beam-up surface, which is normal to the $\hat{y}$ axis and has negative sense.
- `dse` - The downstream surface, which is normal to the $\hat{z}$ axis and has negative sense.
- `append` - Appends an additional surface to the cell description, including the specified sign.

4 Auxiliary components

4.1 Mode cards

The user can specify a mode line near the beginning of the input file in order to change how the importances are written. The options in the `params` entry are either `imptable` (which prints a table at the end of the input) or `impcell` (which prints the importances with the cells). The default is `impcell` if nothing is specified. The particle list can also be given as the `particles` option with a comma separated string that will appear in the resulting MCNP input file.

```
--type mode --params "imptable" --particles "n,p,h,d,t,s,a"
--type mode --params "impcell"
```

4.2 Reading and importing existing MCNP models

The geomwriter input does not need to be a single, monolithic input file. Instead, a primary input file can contain instructions to read another file in order to facilitate process of commenting out a section of the geometry to produce a simpler MCNP input file. The `read` capability that takes a file name as a parameter. The structure of the file follows the same rules as the primary file, with "#" indicating comments. Nested reads (ie. a `read` line inside a file that was `read`) are not supported.

```
--read guide_segments.txt
```

There might be instances where the user wishes to import an existing shielding model and insert a guide network into it. The existing model should be imported near the beginning of the Geomwriter input in order to correctly build the cell/material/surface databases to avoid overwriting them. If the file `MATSimport.in` defines material 7 and the user later tries to define another material 7, Geomwriter will number the new one by checking for the next available number after 7. If instead, the user defines material 7 and then imports `MATSimport.in`, the material 7 in the file will overwrite the user-specified material 7. There are a series of types that facilitate this process:

```
--type materialimport --filename MATSimport.in
--type cellimport --filename CELLSimport.in --ifilename IMPS_import.in
       --omissions "[1051, 1052, 1053, 1054, 1055]"
--type surfaceimport --filename SURFSimport.in
```

For each, Geomwriter processes the file line by line and assigns information to a material/cell/surface until the next such entity is found. Geomwriter determines that a new entity has been found when any of the first 5 characters of a line are not whitespace or a comment, otherwise it is determined that the line is a continuation of the previous one and it continues processing.

For materials, the input expects the following format.

```
c Aluminum
c   density = 2.70 g/cc
c   at.density = 6.04484E-02 a/b-cm
m7    nlib=00c
13027 1
mt7   al27.22t
```

When there is a comment line including the word "density", the script will attach that density to the material inside the script and the material can be used in cells defined in the input file. The input found in the material import file will be echoed to the main output of geomwriter later on.

The cell import file is processed by determining that a new cell is being defined when the beginning few characters of a line contain a number, in the usual MCNP way, and the cell definition can cross multiple input lines. For example, each of the following lines will lead to cell 1 being read into the script. Once a cell has been imported into the script in this manner, that cell can be cut from later cells or later cells can be cut from it. These cells will be echoed to the output of the geomwriter script.

```
1       7 -2.7  -1 2 -3 4 -5 6
        imp:n=1

1       7 -2.7  -1 2
                -3 4
                -5 6
                imp:n=1

1       7 -2.7  -1 2
                -3 4
c               -7 8
                -5 6
                imp:n=1
```

If the importance is specified near the cell input line, then that importance is used. However, there may be instances where the importances are a table in the final block of the input file. In that case, the user should specify an importance file (`ifilename`) on the cell import line so that the importance table can be read and linked with the cells. Additionally, the user may choose to omit certain cells (ie. Geomwriter might replace them) and this can be specified as `omissions` with a list of cell numbers.

Finally, a block of surface definitions can be imported and echoed back to the output of the Geomwriter script. Each surface definition should be included on a single line.

4.3 MCNP material specifications

Materials should be included at the beginning of the input file because they will be used later for cell densities. A material specification line begins with `--type material`. Options include the material number, the density, and the list of ZAIDs.

- `idnum` - the MCNP material card number for this material.
- `density` - the density of the material (negative for g/cm^3, positive for atom density), which will be used for the densities of volumes that use this material.
- a list of ZAIDs and fractions for the composition of the material.

Additionally, the thermal treatment card can be attached to a material on a separate line. The most basic material specification is as follows:

```
--type material --idnum 14 --density -2.33 14028 0.922 14029 0.047 14030 0.031
```

Which produces the following output:

```
m14      14028 0.922
         14029 0.047
         14030 0.031
```

4.3.1 Thermal and SANS material modifications

Additionally, the thermal treatment card can be attached to a material on a separate line. The most basic material specification is as follows:

```
--type thermal --idnum 14 si28ie.98t
```

Which produces the following output:

```
mt14     si28ie.98t
```

The SANS extension in MCNP can be invoked by using the `sans` card, and there are two methods of describing a SANS material. Either the file name for the SANS differential cross section can be supplied using the `fname` card, or the parameters of a hard-sphere model can be included. The sample input cards are as follows:

```
--type sans --idnum 14 --fname sisans.dat
--type sans --idnum 14 --contrast 9.3529e-6 --meandiameter 10.0 --Zschulz 10000.0
--packingfraction 0.1 --pSLD 9.3529e-6 --mSLD 0.0
```

and they produce the following output, respectively:

```
ms14     sisans.dat
ms14     9.3529e-6 10.0 10000.0 0.1 9.3529e-6 0.0
```

4.4 Specifying super mirror parameters

In the case of an IGES guide, the parameters will be read directly from the `.igs` file and it is not necessary to supply the reflective function separately. However, it must be supplied for the `guide` and `guidetaper` components.

The input line is of type `reff` and uses the `idnum` card to specify the ID of the reflection function. The `rflag` option changes the reflective behavior of the guide, with 0 corresponding to default behavior where reflection and transmission happens with probability p and $1 - p$, respectively. A value of 1 corresponds to guaranteed reflection with a weight adjustment corresponding to the reflection probability, that is that the outgoing neutron weight is $w_r = pw_0$. A value of 2 corresponds to neutron banking where a transmitted neutron is created with weight $w_t = (1 - p)w_0$ and a neutron is reflected with weight $w_r = pw_0$. If the `rflag` option is not specified, then a line with `rflag=0` is written to the output, though such a line is not necessary because this is the default behavior of the guide extension. The IGES pre-processor script should write the desired `rflag` option on guide input lines.

The trailing parameters that follow are the definition of the guide reflective function and are in order of R_0 (low-angle reflectivity), Q_c (critical angle), m (the m-value for the material in multiples of Nickel), α (the slope after Q_c), and W (the super-mirror cutoff). The sample input card is as follows:

```
--type reff --idnum 2 --rflag 0 0.99 0.0219 4.0 3.044 0.0025
```

and which produces the following output:

```
reff2      0.990000 0.0219 4.0 3.044 0.0025
rflag2     0
```

4.5 Variables

The `variable` type specifies a variable name and a value and all occurrences of that variable name (with a "$" appended) will be replaced by the value when the input file is parsed. String, floating point, and integer values are supported and a variable can replace any item in a line. Mathematical operations with variables are not supported.

```
--type variable --variable imp1 --value 10
--type variable --variable al27 --value 7
```

Additionally, the surface numbers for every cell are saved as variables internally and can be used for other cell definitions. The format for surface number variables is `$AAAA_n`, where AAAA is the cell number and n is nth surface on the cell definition, counting from 0. In this sense, cells in geomwriter are akin to MCNP macrobodies like `rcc` and `rpp` but utilize an underscore instead of a period. The `specialcase` cell below is defined relative to 5 surfaces from cell 976 and 1 from cell 974 (both of which are unspecified here), with the importance defined as the variable `$imp1`, and with the material defined by variable `al27`.

```
--type specialcase --idnum 978 --material $al27 --imp $imp1 --dse -$974_0
             --ble -$976_0 --bre $976_1 --bde $976_2 --bue -$976_3 --use $976_5
```

The resulting output is as follows with cell 978 defined as negative relative to surface 0 of cell 974 (1012), negative relative to surface 0 of cell 976 (1000), positive relative to surface 1 of cell 976 (1001), positive relative to surface 2 of cell 976 (1002), and negative relative to surface 3 of cell 976 (1003), and positive relative to surface 5 of cell 976 (1005). Cell 978 is comprised of material 7 and has importance 10.

```
  974     1 -7.200000   -1012 +1013 -1014 imp:n=1
  976     0    -1000 +1001 +1002 -1003 +1004 -1005 imp:n=1
  978     7 -2.7    -1000 +1001 +1002 -1003 +1005 -1012 imp:n=10
```

4.6 Tallies

Standard MCNP tallies, tmesh tallies, and fmesh can be specified. As usual, `idnum` specifies the tally id number. There are a number of additional options that specify different aspects of the tally.

4.6.1 Standard MCNP tallies

The tally kind is given as the `--tallykind` option. The tally cell or surface bins (bin), segment divisors (sd), tally segment (fs), energy bins (e), and cosine bins (c) are all specified as comma separated python lists in a string parameter. For example, the following specifies a surface tally across surface 4 of cell 100 (binbounds) and segments it relative to surfaces 0, 1, 2, and 3 of the same cell (fsbounds). Each area is defined to be unity (sdbounds), and it is segmented in energy (ebounds) and angle (cbounds). The first element in the cosine bins list specifies the angle type, with "deg" or "rad" specifying degrees or radians, respectively, with the degrees mode leading to a `*c` card.

```
--type tally --idnum 12 --tallykind "n " --binbounds "[$100_4]"
             --fsbounds "[ +$100_0, -$100_1, +$100_2, -$100_3 ]"
             --sdbounds "[ 1.0, 1.0, 1.0, 1.0, 1.0]"
             --ebounds "[0.00, 1.0e-9, 1e-6, 1e-3, 1]"
             --cbounds "[deg, 90, 20, 10, 5, 2, 1, 0.5, 0.0]"
             --comment "Guide 1 surface tally"
```

This tally specification leads to the following output:

```
f12:n    104
fs12     +100  -101  +102  -103
sd12     1.0 1.0 1.0 1.0 1.0
e12      0.0 1.0e-9 1e-6 1e-3 1
*c12     90 20 10 5 2 1 0.5 0.0
fc12     Guide 1 surface tally
```

Additionally, a tally can be affected by a dose function via de and df cards. This is read in from space separated, text file with the `--dosefile "neutron_dose.txt"` option. The file is expected to be 2-columns with the energy in column 1 and the functional form in column 2.

4.6.2 TMESH mesh tallies

TMESH mesh tallies are specified as the detector type. In this example, the exact contents of the dosefile will be printed at the bottom of the tmesh block, which means that the mshfm cards should be in this file and can be used for multiple tallies. The number of bins (nbins) determines the interpolation for the mesh and only linear interpolation is currently supported. The tallykind will appear on the first line of the (r,s,c)mesh entry, and is treated as plain text. The parameter order in the `binbounds` list is $[a_{\text{low}}, a_{\text{high}}, b_{\text{low}}, b_{\text{high}}, c_{\text{low}}, c_{\text{high}}]$, where a, b, c are the corresponding mesh coordinates.

```
--type detector --idnum 100 --nbins "[640, 0, 400]"
                --binbounds "[-1500, 1500, -40, 40, 600, 2200]"
                --tallykind "p dose 2"
                --dosefile "neutron_gamma_dose.txt"
```

This results in the following output, with the contents of the dosefile removed for brevity.

```
tmesh
rmesh100:p dose 2
cora100   -1500 639i 1500
corb100   -40 40
corc100   600 399i 2200
mshfm2    ...
...
endmd
```

Both spherical (smesh) and cylindrical (cmesh) mesh tallies are implemented via including the word “sphere” or “cylinder” in the `tallykind` parameter. The words “sphere” and “cylinder” will be removed from the tallykind before it is written out. The list specified in `binbounds` will be used for each of these mesh types. Additionally, an energy band can be specified for a tmesh tally via the `ebounds` parameter which looks the same as for standard MCNP tally types above, for example:

```
--ebounds "[1.0e-9, 1e-6]"
```

4.6.3 FMESH mesh tallies

FMESH mesh tallies can also be specified by including the "fmesh" in the card type. Only rectangular tallies are currently supported. Similarly to tmesh, the tallykind parameter will be included in plaintext on the fmesh card line. The following input line:

```
--type detectorfmesh --idnum 104 --nbins "[62, 34, 150]"
                     --binbounds "[-310, 310, -270, 175, 1000, 3500]"
                     --tallykind "n"
```

creates the following rectangular fmesh:

```
fmesh104:n      geom=xyz
         origin -310 -270 1000
         imesh 310
         iints 62
         jmesh 175
         jints 34
         kmesh 3500
         kints 150
```

5 Post-processing

After the input file is processed and before the cells and surfaces are written out, there is a post-processing step that cleans up the resulting MCNP geometry file so that it runs more efficiently. During execution, MCNP will remove redundant surfaces in the problem and this can be quite costly in terms of initialization time because it takes at minimum $O(n^2_{\text{surfaces}})$ time to search all surfaces for redundant surfaces and $O(n_{\text{surfaces}})O(n_{\text{cells}})$ searching for redundant surfaces in cells (and the same for tallies and other quantities). Additionally, supermirror surfaces can be subject to removal for being redundant and the fact that they are supermirrors is not sufficient for regarding them as sufficiently different. Therefore, it is preferred to do this processing before MCNP is forced to do it, though processing currently only addresses planes that do not have accompanying transformation cards. As a result, there may be some remaining redundant surfaces in the geometry, but it will not be a large number since the majority of the geometry script is generating simple planes.

Planes are defined with equations in the form

$$Ax + By + Cz - D = 0, \tag{18}$$

where $\hat{n} = \langle A, B, C \rangle$ is the unit normal to the surface and D is the distance of the plane from the origin along the normal. The surfaces are redundant at the level of $\delta = 10^{-6}$ if the dot product of the unit normal vectors are within δ of unity

$$\hat{n}_1 \cdot \hat{n}_2 > 1 - \delta, \tag{19}$$

and if either fractional difference between the D coefficients is smaller than δ which is checked in this way to avoid dividing by zero.

First, the supermirror surfaces are checked to find all supermirror surfaces that are redundant with other supermirror surfaces. For each redundant supermirror surface, the second version (ie. the later one in surface number) is adjusted along the normal such that it is no longer redundant by choosing a random number, ϵ, from the normal distribution with $\mu = 0$ and $\sigma = 10^{-4}$

$$D_{2,\text{new}} = D_2 + \epsilon, \tag{20}$$

such that the new plane equation for surface 2 is

$$A_2x + B_2y + C_2z - D_{2,\text{new}} = 0. \tag{21}$$

The list of mirrors is checked again for redundant supermirrors if any were found, to ensure that no redundant mirror was moved and made redundant with a different one. An output file is produced for this process, named with the base input file name plus “redundantM” appended to the end of it, that contains a list of the surface definitions of supermirror surfaces along with a list of those that were changed:

```
Master 30744  p     0.000000 1.000000 -0.000000 1.499644
  Old  31512  p     0.000000 1.000000 -0.000000 1.499643
  New  31512 p    0.0 1.0 -0.0 1.4996527405539881
  dict 31512 p    0.0 1.0 -0.0 1.4996527405539881
```

where the first line main surface (30744) with its surface definition, the second line is the same information for the redundant surface (here differing in D coefficient by 10^{-6}) and the third line is the new surface definition for the redundant cell (now differing by 9.7×10^{-6}).

Next, the non-supermirror plane surfaces are checked to find all redundant planes and a similar output file is created named "redundantNM". If a plane is found to be redundant to another, the first one in surface number is determined to be the "master" surface and will also be the master surface for additional surfaces that are redundant to the same surface. The redundant copies are appended to a list of surfaces to be deleted from the surface dictionary, and the following lines are appended to "redundantNM" log file:

```
974 [980, 986, 992, 1019]
980 974  p     0.000000 0.000000 1.000000 295.092000
986 974  p     0.000000 0.000000 1.000000 295.092000
992 974  p     0.000000 0.000000 1.000000 295.092000
1019 974  p     -0.000000 -0.000000 1.000000 295.092000
```

where surfaces 980, 986, 992, and 1010 were flagged as the same as 974 and 974 will be used as the master surface while the other 4 will be deleted, and the surface definitions of the surfaces that will be deleted are listed. Following this step, all cell definitions are checked for the inclusion of surfaces that have been deleted and those occurrences are replaced by the appropriate master surface. The following original and new cell definitions are appended to the log file:

```
orig 971  -976 +977 +978 -979 +980 -981
     #( -970 +971 +972 -973 +974 -975)
new 971 -976 +977 +978 -979 +974 -975
     #( -970 +971 +972 -973 +974 -975 )
```

where surfaces 980 and 981 have been replaced by 974 and 975, respectively. The following comments are appended to the cell 971 comment string:

```
c replace surface 980 with 974
c replace surface 981 with 975
```

which appears in the resulting MCNP input geometry file. A count of the number of original non-mirror surfaces is reported, as well as the number of redundant non-mirrors, and the new number of surfaces after redundant surfaces are deleted. For a relatively small HFIR Cold Guide Hall model, these numbers are 1747, 483, and 1264, respectively.

6 IGES Pre-processor

The McStas model of a neutron beamline can be quite complex, including velocity selectors and hundreds of guide segments. These are contained within a zip repository for ease of sharing the model from McStas to both CAD and the MCNP geomwriter script, but it is cumbersome to produce the geomwriter input by hand for a zip repository with hundreds of `.igs` files which makes an accompanying pre-processor necessary called `iges_read.py`. This script is where some facility or beamline dependent choices can be made, including how to model velocity selectors or the exact geometry of the glass, vacuum jacket, and casing. The script takes 4 inputs

```
python3 iges_read.py NBX\_20201105 0.0 50000 NBX > ID_IGES.txt
```

1. repository file name: can be a zip or standalone file (NBX_20201105.zip)
2. delta: a floating point number to move velocity selector shielding (0.0)
3. Initial cell/surface number (50000)
4. The beamline name identifier (NBX)/

The NBX folder should be created before running the script. It serves as a storage location for any `.off` files generated by the script in order to keep files organized.

This script expects the `zip` repository to contain a sub-folder and to have a `.log` file, as well as a series of subtype files that serve as file type classifiers. As an example (where NBX is the alpha-numeric code, DATE is the production date for the file, and N is the subtype):

- `NBX_TOP_LEVEL_DATE_IGES.log` - A log file from the production of the IGES repository, and which connects `.igs` file names to McStas component types and component names in the orignal instrument file.
  - The following line signifies that `NBX_guide0172.igs` is a `Guide_channeled_tally` component and was named `NBX_Guide1_03` in the original instrument file.
  - `172 NBX_guide0172 Guide_channeled_tally NBX_Guide1_03`
- `NBX_TOP_LEVEL_DATE_IGES.igs` - Instructs the parser which subtype files are to be read.
- `NBX_subtype_N_DATE.igs` - where N is a tag for the engineering system. These files contain pointers to individual component `.igs` files of each subtype.

This numbering scheme allows the pre-processor to easily distinguish between neutron guide IGES files and velocity selectors or monochromators. For ease of processing, some similar kind of numbering scheme should be used.

The pre-processor searches for the `.log` file and the subtype files (currently only optical components) and then processes the `.log` file to produce a dictionary of component IDs, `.igs` file names, corresponding McStas component types, and McStas instrument names. Then the optical subtype files are processed, producing a dictionary of the the `.igs` files that will be read and which is connected to the information from the `.log` file. Following this step, the dictionary is processed in order by the component ID and the exact method that it will be processed depends on the McStas component type. In general, information about the McStas component name are used as comments in the MCNP input file to facilitate human readability of the output.

The global section of the IGES file standard [1] contains information about the production of the file, author, and parameters for reading the file. Item 14 (starting from 1) contains a numerical flag indicating the unit scale of the file, and this converted to MCNP units (centimeters) if necessary.

6.1 Rectangular Guides

The internal guide volume is defined by the geometry given in the IGES subtype file. This internal guide volume is used to generate 3 more volumes, namely the guide glass, a vacuum jacket, and a steel casing via an expansion algorithm using pre-defined expansion amounts. Very close to the shutter (under 2 m), these thicknesses are 1 mm, 1.1 mm, and 1.2 mm in order to prevent overlaps between beamlines. From 2 m to 10 m, these values are 2 mm, 2.1 mm, and 2.2 mm, respectively (again, to prevent overlaps). After 10 m, realistic thicknesses are used (6 mm, 10 mm, and 13 mm). The expansion algorithm takes all planes except for the end-caps and shifts them along the surface normal vectors by the expansion amount and then determines the new point boundaries of the volume by checking for the intersections of all transverse planes with the end planes (which are preferred to be given in the IGES file).

After these have been determined, the `iges_read.py` script prints an igeselement line that can be read by `geomwriter.py` for the internal volume and the expanded volumes. The proper cut volumes are also generated from the positions of the end faces of the segment and a ruled grid of enclosure cells, but some additional care must be taken for penetrations through shielding through graphical checks and modifying a routine in the `iges_read.py` script. Examples are shown below:

```
--type igeselement --igeszip "NB1_IMAGINE_20201105.zip"
       --igessubfile "NB1_IMAGINE_IGES/NB1_guide0170_20201105.igs"
       --comment "NB1_Guide1_01 20201105.163525" --idnum 50000
       --idnumoffset 50000 --imp $imp1 --cutfrom 50001
       --material 0 --stddev 0.001 --mirror 1
--type igeselement --comment "NB1_Guide1_01 20201105.163525 glass"
       --igeszip "NB1_IMAGINE_20201105.zip"
       --igessubfile "NB1_IMAGINE_IGES/NB1_guide0170_20201105.igs"
       --idnum 50001 --idnumoffset 50001 --imp $imp1 --cutfrom 50002
       --material 13 --expand 0.1 --mirror 0
--type igeselement --comment "NB1_Guide1_01 20201105.163525 gap vacuum"
       --igeszip "NB1_IMAGINE_20201105.zip"
       --igessubfile "NB1_IMAGINE_IGES/NB1_guide0170_20201105.igs"
       --idnum 50002 --idnumoffset 50002 --imp $imp1 --cutfrom 50003
       --material 0 --expand 0.11 --mirror 0
--type igeselement --comment "NB1_Guide1_01 20201105.163525 casing"
       --idnum 50003 --igeszip "NB1_IMAGINE_20201105.zip"
       --igessubfile "NB1_IMAGINE_IGES/NB1_guide0170_20201105.igs"
       --idnumoffset 50003 --imp $imp1 --material 1 --expand 0.12
       --cutfrom 9003 --cutfrom 9002 --cutfrom 9003 --mirror 0
```

Each of the outer volumes will use the same base `.igs` file in the `geomwriter.py` script, and will be expanded by the given amount by the same routines as were used in `iges_read.py`. These volumes also omit supermirror properties (`--mirror 0`). The script then continues and processes the next `.igs` file in the same manner. Once two such files have been processed, the script checks the distance between the end caps of each. If this distance is under 1 mm, the script must stitch together these guide segments with a vacuum volume and a continuous steel casing because a gap would likely be comprised of air in the MCNP geometry and would cause incorrect results.

The four cells specified above that comprise the guide, glass, vacuum jacket, and casing appear in the output as follows along with some automatically generated comment lines to provide some detail about what the script is doing:

```
c cccccccccccccccccc
c NB1_IMAGINE_IGES/NB1_guide0170_20210430.igs NB1_Guide1_01
50000     0   -50000 +50001 -50002 +50003 +50004 -50005
          imp:n=170
c cccccccccccccccccc
c NB1_IMAGINE_IGES/NB1_guide0170_20210430.igs NB1_Guide1_01  glass
c cut cell 50000
c replace 50010 with 50004
c replace 50011 with 50005
50001    13 -2.200000   -50006 +50007 -50008 +50009 +50004 -50005
         #( -50000 +50001 -50002 +50003 +50004 -50005 )
          imp:n=170
c cccccccccccccccccc
c NB1_IMAGINE_IGES/NB1_guide0170_20210430.igs NB1_Guide1_01  gap vacuum
c cut cell 50001
c replace 50016 with 50004
c replace 50017 with 50005
c replace 50010 with 50004
c replace 50011 with 50005
50002     0   -50012 +50013 -50014 +50015 +50004 -50005
         #( -50006 +50007 -50008 +50009 +50004 -50005 )
          imp:n=170
c cccccccccccccccccc
c NB1_IMAGINE_IGES/NB1_guide0170_20210430.igs NB1_Guide1_01  casing
c cut cell 50002
c replace 50022 with 50004
c replace 50023 with 50005
c replace 50016 with 50004
c replace 50017 with 50005
50003     1 -7.200000   -50018 +50019 -50020 +50021 +50004 -50005
         #( -50012 +50013 -50014 +50015 +50004 -50005 )
          imp:n=170
c cccccccccccccccccc
```

In several instances, the cell definition and surface list have been cleaned up in post-processing by removing redundant surfaces and this is evident in the comments that surface 50005 has replaced surface 50017, for example:

```
c replace 50017 with 50005
```

The surfaces that appear in the cell definitions above are shown below. Several correspond to the same face on other cells, such as 50004 and 50005 which are the upstream and downstream surfaces, respectively. Additionally, surfaces 50000, 50006, 50012, and 50018 are the beam up surfaces of each cell, with 50000 being the upper face of the guide and the following three expanded outward from 50000 by 0.0493 cm, 0.0993 cm, and 0.1493 cm, respectively.

```
50000  p    0.000000 0.999871 -0.016052 2.695538
50001  p    0.000000 0.999871 -0.016052 -1.303791
50002  p    0.999933 0.000185 0.011535 2.138890
50003  p    0.999933 0.000185 0.011535 -1.861525
50004  p    -0.011537 0.016051 0.999805 482.053746
50005  p    -0.011537 0.016051 0.999805 534.828266
50006  p    0.000000 0.999871 -0.016052 2.744854
50007  p    0.000000 0.999871 -0.016052 -1.355146
50008  p    0.999933 0.000185 0.011535 2.188119
50009  p    0.999933 0.000185 0.011535 -1.911879
50012  p    0.000000 0.999871 -0.016052 2.794854
50013  p    0.000000 0.999871 -0.016052 -1.405146
50014  p    0.999933 0.000185 0.011535 2.238119
50015  p    0.999933 0.000185 0.011535 -1.961879
50018  p    0.000000 0.999871 -0.016052 2.844854
50019  p    0.000000 0.999871 -0.016052 -1.455146
50020  p    0.999933 0.000185 0.011535 2.288119
50021  p    0.999933 0.000185 0.011535 -2.011879
```

The stitching regions are closed by producing a series of `.off` files. In order to produce these stitching regions, the script takes the downstream end of the previous segment and the upstream end of the next segment and processes these points with the SciPy ConvexHull algorithm to find all triangular facets. These are then reduced in number by checking for surfaces that have the same sense relative to the center of the segment and which are nearly co-planar ($\hat{n}_1 \cdot \hat{n}_2 > 0.99$). Then, all intersections of all planes are found and checked to ensure that no additional points have been introduced by ensuring that no point is too far from the center of the volume and that no point is too close to another. If this process fails because the stitching region is too small, the process is re-attempted after stretching the planes of the region farther apart and then re-positioned correctly before writing them. Once the list of corners has been established, an off file is written with a corresponding input line appended to the pre-processor output. Additionally, the downstream (upstream) end surfaces of the previous (next) guide segments are appended to the definition.

```
--type offile --offile NB1/0_NB1_guide0171_20210430.off --idnum 50008 --material 0
         --anchor 0 --scalefactor 1.0 --idnumoffset 50008 --cutfrom 50009 --imp $imp1
         --skipredundancy 1 --append -$50004_4 --append $50000_5
         --comment "NB1_Guide1_02 20201105.163525
         NB1/0_NB1_guide0171_20201105.off connector"
```

The stitching region can be quite small and an example is shown in figure 8, which shows the inner stitching cell numbered 50048.

The corresponding stitching cell and surfaces are as follows:

```
c NB1_IMAGINE_IGES/NB1_guide0171_20210430.igs NB1_Guide1_02
c NB1/0_NB1_guide0171_20210430.off connector
c surface modification appending ['-$50004_4', 50005]
c replace 50050 with 50005
c replace 50053 with 50028
50008     0    +50048 -50049 +50005 -50051 +50052 -50028 -50028
               +50005
          imp:n=171
c cccccccccccccccccc
```

The `.off` file that was written by the pre-processor is as follows:

```
OFF
# stuff
#NB1/0_NB1_guide0171_20210430.off
# convex hull with 12 triangular simplices
#      should have 6.0 faces
8 6 0
-4.03206702 7.279997149999994 534.7693835699997
-8.031799020000003 7.279256399999998 534.7232431700002
-4.032644 7.280798369999999 534.8191745999998
-8.032376000000003 7.280057620002581 534.7730342001605
-4.032067019999998 11.279481149999986 534.7051747699999
-8.031799019999996 11.278740399999995 534.6590343700003
-4.032644000000001 11.280282379997411 534.7549657998396
-8.032376000000005 11.279541630000002 534.7088254000001
# surfaces
4 0 1 2 3
4 4 5 6 7
4 0 1 4 5
4 0 2 4 6
4 1 3 5 7
4 2 3 6 7
```

The MCNP input file then contains the following:

```
50005  p    -0.011537 0.016051 0.999805 534.828266
50048  p    0.000000 0.999871 -0.016090 -1.325154
50049  p    0.000000 0.999871 -0.016090 2.674738
50051  p    0.999933 0.000186 0.011584 2.164465
50052  p    0.999933 0.000186 0.011584 -1.835533
50028  p    -0.011537 0.016051 0.999805 534.878067
```

An example of three guide segments 50028, 50036, and 50044 is shown along with the glass, vacuum jackets, and casings for those segments in figure 6. The thin regions that look like thick black lines are the thin stitching regions similar to cell 50008 above. A vertical cross section of a guide segment with the glass, vacuum jacket, and casing cells is shown in figure 7.

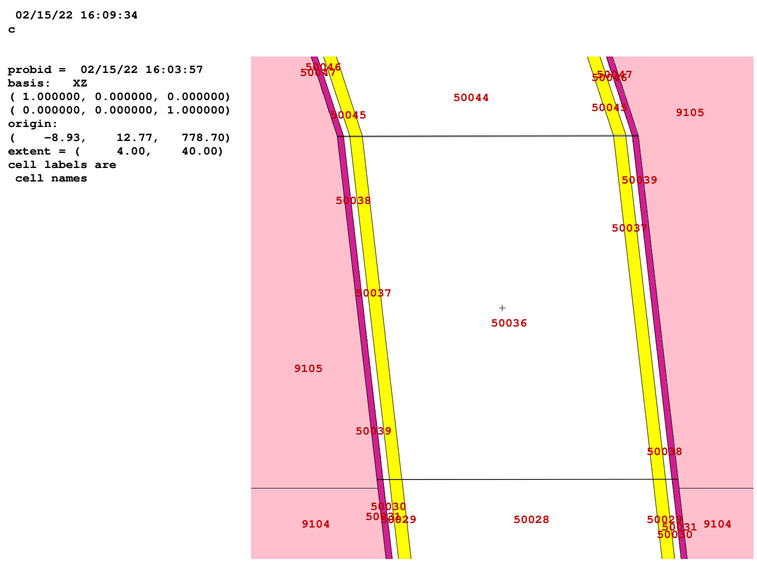

Figure 6. Rectangular guide segments at constant height in the x-z plane. Neutrons are transported from the bottom with reflecting surfaces to the left and right (top and bottom are not shown). The vertical direction is compressed by a factor of 10 compared to the horizontal (80cm vs 8cm) in order to show more detail.

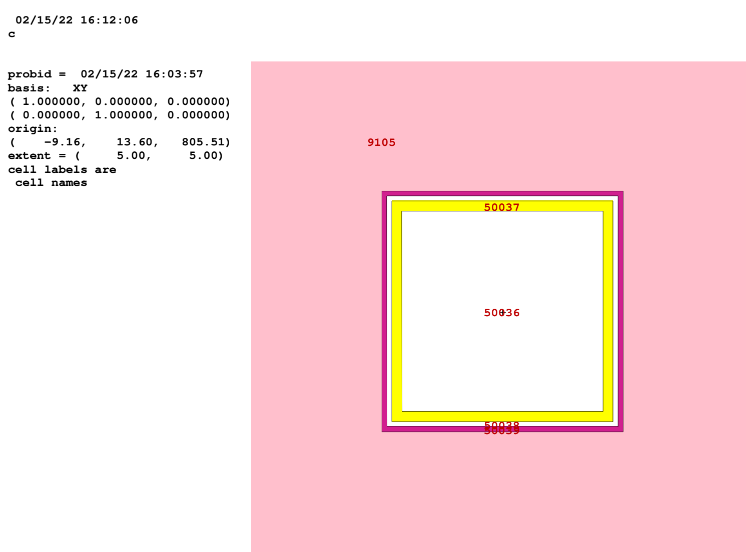

Figure 7. The same guide segment (50036) in the x-y plane, showing all 4 reflecting surfaces as well as the glass (50037), vacuum jacket (50038), external casing (50039), and the surrounding air volume (9105).

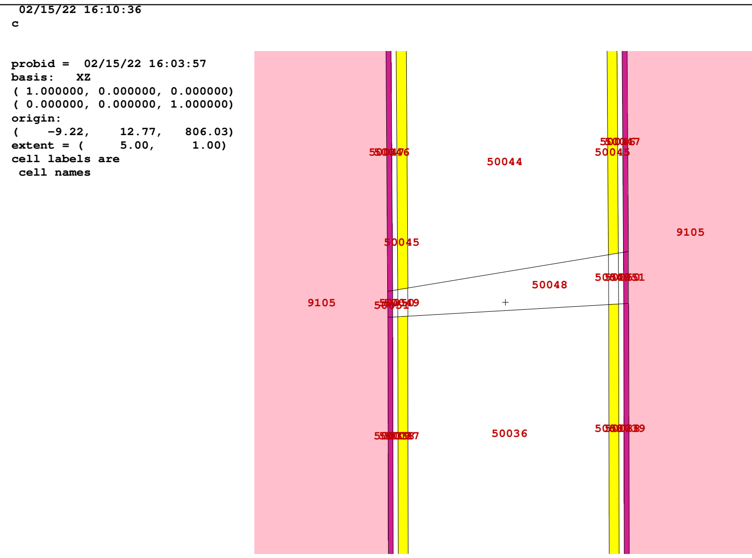

Figure 8. The stitching region (50048) between two consecutive guide segments (50036 and 50044), which was described using an object file format based on the upstream and downstream ends of the adjacent guide segments.

6.2 Channeled guides

Channeled guides are those with multiple parallel internal channels separated by thin septa, and is differentiated from the standard rectangular guides by both McStas type ("channeled") and by the number of boundaries in the IGES file not being either 4 or 6. The pre-processing of such a file depends on the format being static such that the surface order always matches fig. 9.

In the example of a neutron guide with 3 channels (left, center, and right), there are 10 surfaces defining the interior guide volume and the pre-processor expects a specific surface order(see figure 9). The guide is then assembled out of the following 4 cells:

- Surfaces [0, 1, 2, 3, 4, 5] comprise the outer boundary of the guide segment, and this cell will be glass and is used to expand for the glass substrate, vacuum, and casing and the guide properties are ignored.
- Surfaces [0, 1, 9, 2, 4, 5], [0, 1, 6, 7, 4, 5], [0, 1, 3, 8, 4, 5] comprise the inner volume of each of the guide channels. These cells will be cut from the first and are vacuum and have guide properties.
- The septa are the regions between surfaces 7 and 8 and surfaces 6 and 9.

The lists of subset surfaces will then be indicated as `igessegment` options and only those subsets of each IGES file are used by the `geomwriter` script to produce each individual cell. The first cell will comprise only the septa between channels after the channels are cut from it. The boundaries of the first cell are also involved in determining the vacuum stitching region between guide cells.

An example input set for a 3 channel guide is shown below. It is quite cumbersome to write manually generate such a configuration and it is instead handled entirely by the pre-processor. The pre-processor does, however, rely on the fact that the list of boundaries in the iges file has a known order, which makes it

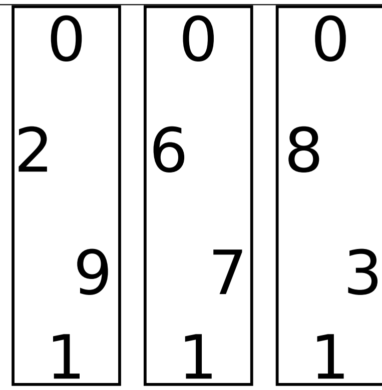

Figure 9. The surface numbering pattern for a 3 channel guide system, with surface 4 (upstream endcap) and surface 5 (downstream end cap) omitted.

straight forward to determine the appropriate `igessegment` set.

```
--type igeselement --igeszip "MANTA2_IGES20210917.zip"
          --igessubfile "MANTA_guide0297_20210916.igs"
          --comment "NB6_MANTA_guide0297_20210916.igs NeuB6_Guide1_132 "
          --idnum 71568 --idnumoffset 71568 --imp 297 --volume 623.5375028956629
          --cutfrom 71569 --material 14 --mirror 0
          --igessegment "[75, 61, 19, 33, 47, 89]" --igesorderid 62970
--type igeselement  --igeszip "MANTA2_IGES20210917.zip"
          --igessubfile "MANTA_guide0297_20210916.igs"
          --comment "NB6_MANTA_guide0297_20210916.igs NeuB6_Guide1_132  glass"
          --idnum 71569 --idnumoffset 71569 --imp 297 --volume 226.6231368634053
          --cutfrom 71570 --material 13 --expand 0.6
          --igessegment "[75, 61, 19, 33, 47, 89]" --mirror 0
--type igeselement --igeszip "MANTA2_IGES20210917.zip"
          --igessubfile "MANTA_guide0297_20210916.igs" -
          --comment "NB6_MANTA_guide0297_20210916.igs NeuB6_Guide1_132  gap vacuum"
          --idnum 71570 --idnumoffset 71570 --imp 297 --volume 167.06961436129097
          --cutfrom 71571 --material 0 --expand 1.0
          --igessegment "[75, 61, 19, 33, 47, 89]" --mirror 0
--type igeselement --igeszip "MANTA2_IGES20210917.zip"
          --igessubfile "MANTA_guide0297_20210916.igs"
          --comment "NB6_MANTA_guide0297_20210916.igs NeuB6_Guide1_132  casing"
          --idnum 71571  --idnumoffset 71571 --imp 297 --volume 133.69566032821263
          --material 1 --expand 1.3  --cutfrom 9008 --mirror 0
          --igessegment "[75, 61, 19, 33, 47, 89]"
--type igeselement --igeszip "MANTA2_IGES20210917.zip"
          --igessubfile "MANTA_guide0297_20210916.igs"
          --comment "MANTA_guide0297_20210916.igs NeuB6_Guide1_132  channel"
          --idnum 71572 --idnumoffset 71572 --imp 297 --volume 623.5375028956629
          --cutfrom 71569 --cutfrom 71568 --material 0 --mirror 1 --rflag 0
```

```
          --stddev 0.001 --igesorderid 62970
          --igessegment "[145, 47, 19, 33, 75, 89]"
--type igeselement --igeszip "MANTA2_IGES20210917.zip"
          --igessubfile "MANTA_guide0297_20210916.igs"
          --comment "MANTA_guide0297_20210916.igs NeuB6_Guide1_132  channel"
          --idnum 71573 --idnumoffset 71573 --imp 297 --volume 623.5375028956629
          --cutfrom 71569 --cutfrom 71568 --material 0 --mirror 1 --rflag 0
          --stddev 0.001 --igesorderid 62970
          --igessegment "[117, 103, 19, 33, 75, 89]"
--type igeselement --igeszip "MANTA2_IGES20210917.zip"
          --igessubfile "MANTA_guide0297_20210916.igs"
          --comment "MANTA_guide0297_20210916.igs NeuB6_Guide1_132  channel"
          --idnum 71574 --idnumoffset 71574 --imp 297 --volume 623.5375028956629
          --cutfrom 71569 --cutfrom 71568 --material 0 --mirror 1 --rflag 0
          --stddev 0.001 --igesorderid 62970
          --igessegment "[61, 131, 19, 33, 75, 89]"
```

The pre-processor generates 7 cells per guide segment for a 3 channel guide. The first 4 are analogous to the standard guide implementation with no channels. The first cell is the interior boundaries of the guide (71568) and which is comprised of silicon rather than vacuum because the guide channels will be cut from it and it will contain the septa between the channels. The next 3 are the glass layer (71569), vacuum (71570), and steel casing (71571). The next 3 are the supermirror guide channels (71572 - 71574) and are cut from the silicon cell (71568). THe resulting geometry is shown in X-Y cross section in figure 10.

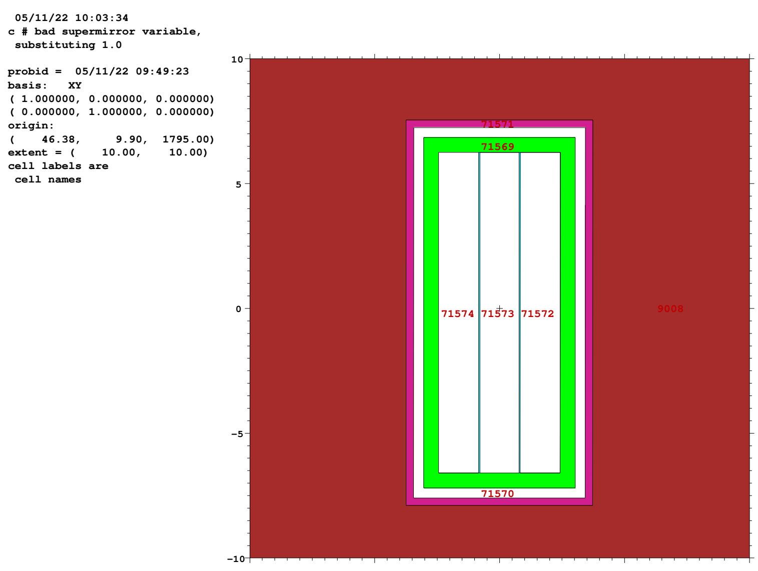

Figure 10. The MCNP geometry of a guide segment with 3 channels. The lines separating cells 71572, 71573, and 71574 are the septa and are comprised of silicon. Glass (green) surrounds the channels, followed by the vacuum jacket, steel casing (pink), and air (maroon).

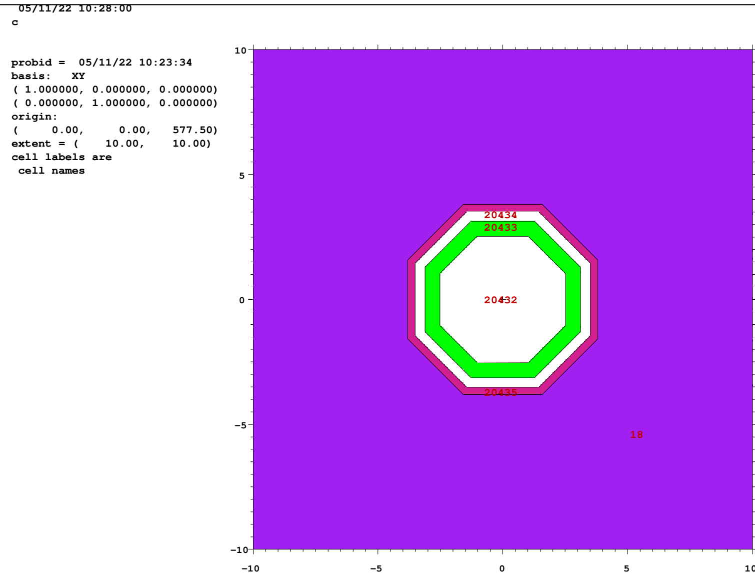

Figure 11. The MCNP geometry of an octagonal guide segment. The inner reflecting volume (20432) is used in an analogous way as for rectangular guides for expansion for the the glass (green), vacuum jacket, and steel casing (pink). The external air is purple.

6.3 Non-rectangular guides

Octagonal and other specially shaped guides must be handled with another special case by a facility dependent pre-processor step. One such case that has been developed in the pre-processor is a guide system with an octagonal cross section. The `igessegment` functionality is used to construct this geometry and only the input line for the interior is shown here:

```
--type igeselement --igeszip "CHESS_Jan_2022_IGES.zip"
        --igessubfile "Zeppelin6_2022_01_01.igs" --comment "Guide6_2022_01_01 "
        --idnum 20432 --idnumoffset 20432 --imp 8 --volume 104.34891960220438
        --cutfrom 20433 --material 0 --mirror 1 --rflag 0 --stddev 0.001
        --igessegment "[17, 33, 49, 65, 81, 97, 113, 129]" --igesorderid 20080
```

The resulting geometry for this segment is shown in figure 11. In general, `igessegment` is able to handle arbitrary polygonal files once the surface ordering has been determined.

6.4 Velocity Selectors

During the pre-processing step in the `iges_read.py` script, a generic velocity selector shielding box is assembled from a series of `.off` files and placed at the location indicated in the IGES repository. Additionally, a rotating velocity selector wheel is placed inside the shielding at the location specified in the IGES repository.

The figure 12 shows an example of a velocity selector shielding package that is placed at a location given by the velocity selector object in the IGES representation of a beamline. The chipper wheel is 35237 and is located at the velocity selector position from the IGES. Adjacent to it are two beam blocks that get placed automatically to facilitate calculations but that can be replaced with air manually if required. There is then a nested shell of shielding surrounding the velocity selector. The shielding box is comprised (from inside to outside) of a boron carbide layer, a structural steel layer, multiple layers of lead, and a structural steel layer. These layers are assembled from an `.off` file that is expanded in all directions to create the layering and to allow for different materials to be placed in different layers.

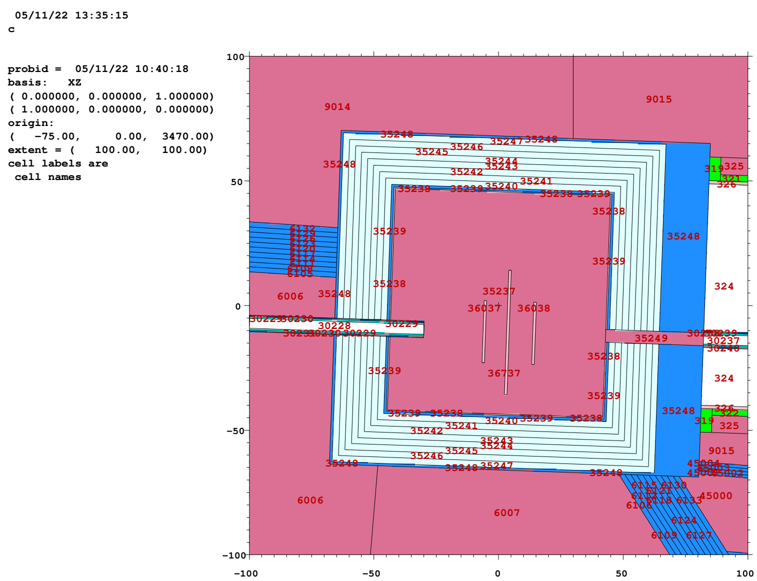

Figure 12. The MCNP geometry of a velocity selector shielding box. Air is pink, steel is darker blue, lead is light cyan.

7 Acquiring and using the software

The routines are developed as a set of python3 scripts which can be recovered from the internal ORNL git repository `git@code.ornl.gov:sns-neutronics/mcnpgeometry.git`. The files are as follows:

- `geomwriter.py` - the primary script for producing an MCNP CGS geometry
- `iges_read.py` - the facility dependent IGES pre-processor
- `snspycommon.py` - a set of common routines used by geomwriter| and `iges_read`
- `recover_input.py` - a script for recovering an MCNP input from the standard MCNP output file
- `off_expander.py` - a script for expanding an `off` file to create an onion-like structure
- `skeleton_tester.py` - a script for pre-processing a large IGES file to create a series of convex hulls
- `mcnp_source_gen.py` - produces a neutron sdef file from an angular binned type 2 MCNP tally

There are a number of python3 dependencies including geomdl, numpy [5], SciPy [8], and Sklearn [7].

The `geomwriter.py` script takes two parameters as input on the command line. The `-i` parameter is the input file that should be read and the `-c` parameter is a common file that will be appended to the output at the end of the script. The common file can contain the source definition that MCNP should use as well as tallies and the calculation mode particles, or anything else that would be in the last input block of an MCNP input file. The output of the script is written to the stdout stream and can be redirected to a file.

```
python3 geomwriter.py -i facility.txt -c common.txt > facility.in
```

The script parses the lines of the input file and generates a set of dictionaries containing cell definitions, surface definitions, transformations, and other appropriate dictionaries containing subsets of the output information. After the complete input file has been processed, the script then loops through the cell dictionary and prints all cells in numerical order and takes some care to ensure that only 80 characters are on each line and a line is skipped at the end of the cell output. The surface dictionary is then used to write all surfaces in numerical order and another line is skipped at the end. The material dictionary is then processed in numerical order and thermal/SANS treatments added where applicable. Transformation cards are printed next in a format with the linear translation on line 1, followed by the 3x3 rotation matrix on the next 3 lines. The cell importances are then printed in a large block (rather than being attached to the cell section) with 10 importances per line in the numerical order of the cells. Finally, the common file is appended to the end.

7.1 General work flow

The geometry of a single beam line, and more so the geometry of a facility, will be quite large with thousands of cells and surfaces. Each cell must be "cut" from those that it overlaps. A facility likely has transport tunnels where multiple guide segments pass quite closely to each other, and the first order attempt might be to have a single, large section of air that is perhaps 10 m long for such a tunnel. Each guide segment that passes through this tunnel must be cut from the tunnel. Unfortunately, MCNP has a limit on the complexity of a cell definition at approximately 2000 "words", where each surface that is part of the cell definition is a "word". For this reason, a good starting point for importing a guide model is an outer grid of cells spanning the entire project geometry, constituting “the building”. These should be assembled as regularly spaced cells numbered in order of distance from the origin, because other cells (namely guide casings) will be cut from these cells based on the position of the centroids of the guide end caps. From this point, an IGES guide model can be processed by the `iges_read.py` script (which may require facility dependent modifications), processed through `geomwriter.py`, and the result will be an MCNP geometry that requires minimal clean-up before use in transport calculations.

Additionally, the shielding structures can be imported using IGES, `.off`, or simple geometric structures and built into a static shielding model such that different guide models can be imported into that model as needed. Instrument specific input should be included in separate input files with `read` lines in the main input file, which allows for the generation of modular MCNP geometries that include a subset of the beamlines for the facility for faster calculation times or for faster visual inspection with the MCNP plotter.

If Geomwriter-built geometry is to be merged with an existing MCNP model, the `import` lines for that model should precede any other materials, surfaces, or cells in the Geomwriter input in order to avoid overwriting anything. If necessary, users can create their own pre

8 HFIR Cold Guide Hall Example

The HFIR Cold Guide Hall has been constructed using the `geomwriter.py` and `iges_read.py` scripts. The shielding has been built manually from (x, y, z) point locations as `.off` files that have been processed with an expansion algorithm in order to create "onion" layers for variance reduction purposes in MCNP. The full shielding model for the HB4 tunnel and transition building(figs. 15 and 16) and the common shielding (figs. 17 and 18) is shown in figure 13. The workflow allows for easy modification of the primary input file to include a subset of the guides in the model simply by uncommenting the `read` lines in the input, and an example includes 14. These figures shows the regular grid structure along the beam direction, and these cells are the ones that shielding and other components are "cut" from. This grid is used because there is a limit to the length of a cell definition in MCNP. The exterior shape of the regular grid was modified to match the boundaries from an `.off` file constituting the boundaries of the building by using parameters like `--ble`.

The HB4 tunnel and transition building (fig. 15) are meshed together by using the various surfaces of each structure to create step-pyramid-like structure in the first bulkhead (fig. 15), which allows for better importance stepping from the inside to the outside. The same step-pyramid-like structure was used at the second bulkhead where the common shielding meets the bulkhead (fig. 17). The onion layers also extend into the floor (figs. 16 and 18).

A model in which the guides were truncated in order to limit the number of surfaces and facilitate graphical loading and inspection is shown in figures 19 and 20. Each guide model was imported from a `.zip` repository containing the `.igs` model that has been converted from the original McStas instrument file. This facilitates quick updates to the MCNP model even after substantial changes to the McStas instrument and beamline specific structures can be commented out in order to produce an model containing any combination of beamlines. For this truncated model, the original number of output surfaces was 10241 and this was reduced to 6677 by the redundant surface post-processing step, producing a model that is quicker to load at MCNP runtime and easier to handle graphically for inspection.

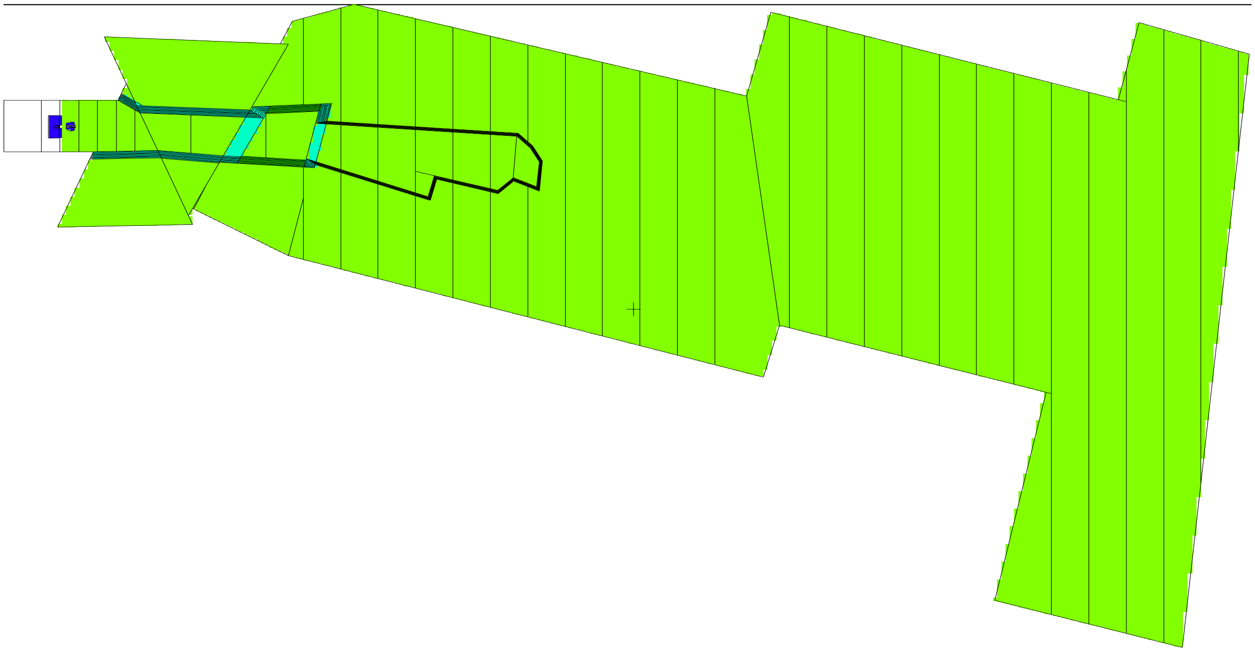

Figure 13. Shielding model for the HFIR Cold Guide Hall. Neutrons begin from the left and pass through the shutter, HB4 tunnel, 1st bulkhead (light blue), transition building, second bulkhead (light blue), and then enter the common shielding area. Air is green.

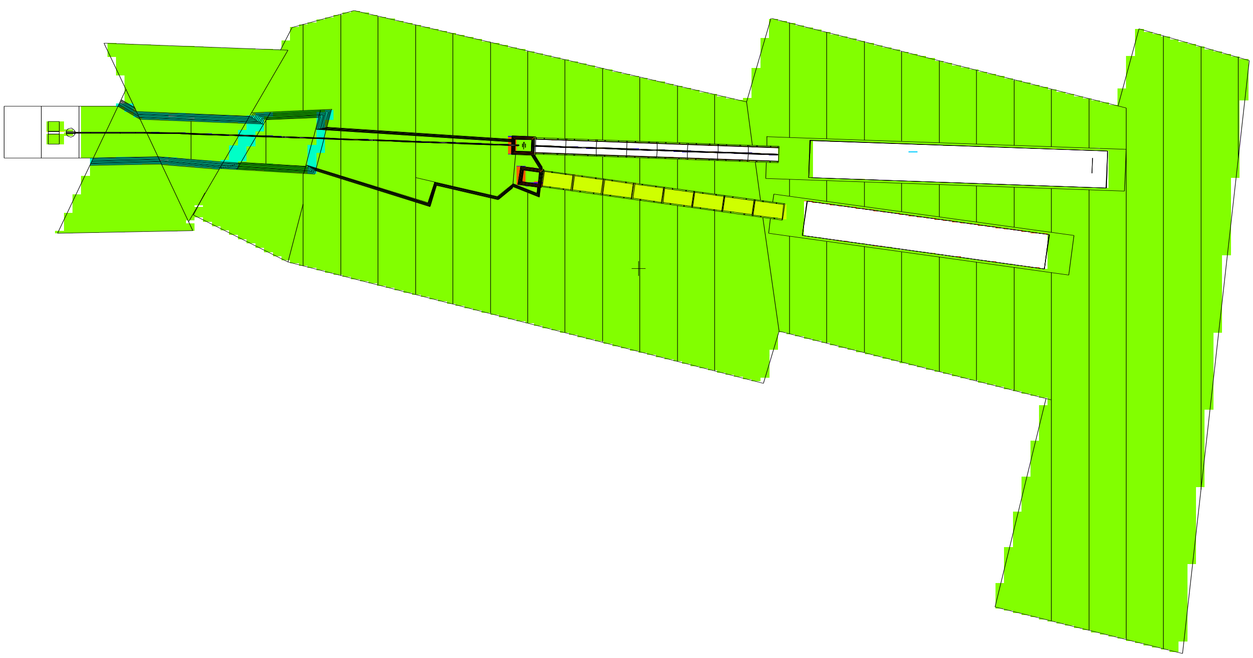

Figure 14. Complete path of the NB5 model (GP SANS). This model is a joint model of NB3 and NB5, and thus shows both SANS tanks, velocity selector shielding boxes, and collimator boxes.

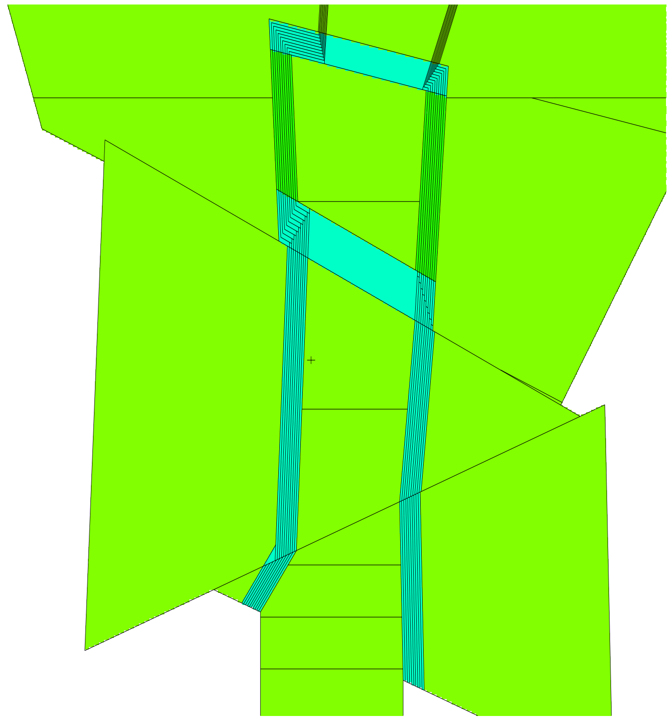

Figure 15. The HB4 tunnel region (lower section) and transition building shielding (upper section) depicting the shielding onion layers.

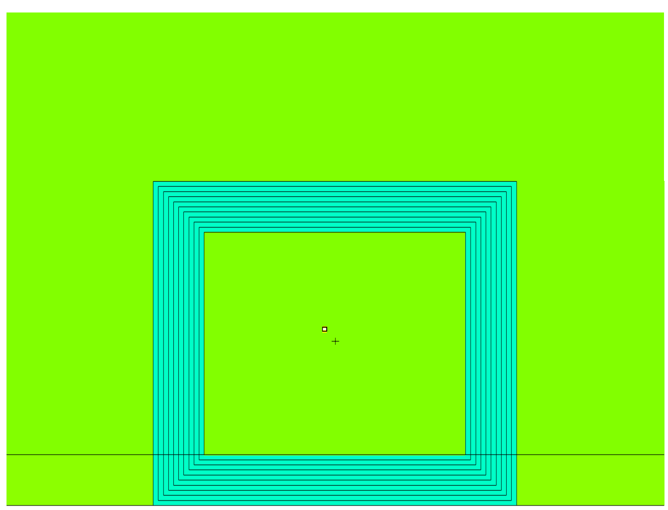

Figure 16. The HB4 tunnel shielding depicted in the x-y plane showing the onion layers extending into the floor, with one beamline (NB1) passing through it.

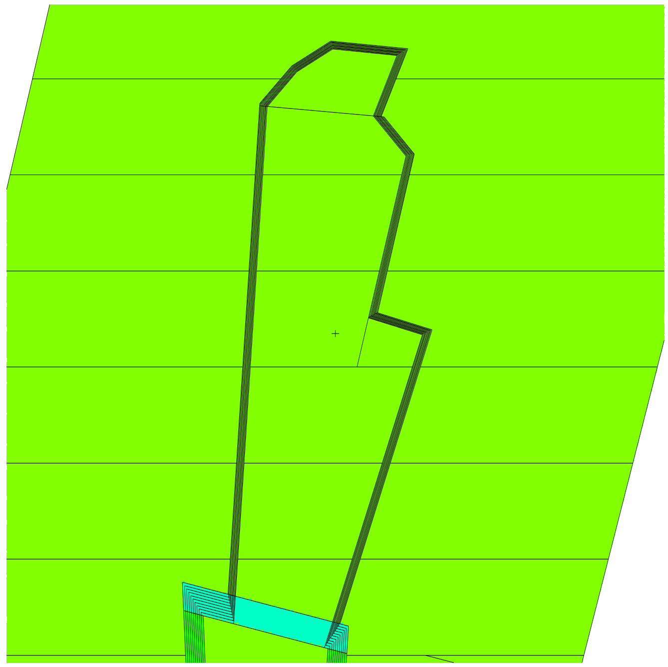

Figure 17. The common shielding region inside the HFIR Cold Guide Hall showing the step-pyramid-like connection at the second bulkdhead.

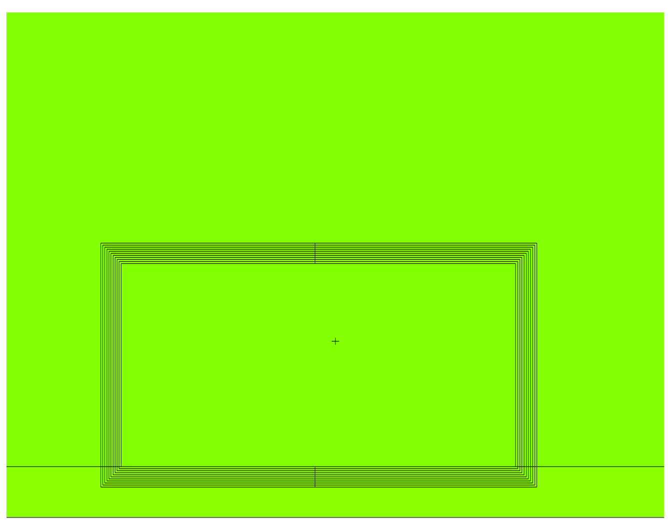

Figure 18. The common shielding in the x-y plane showing the onion structed extending into the floor.

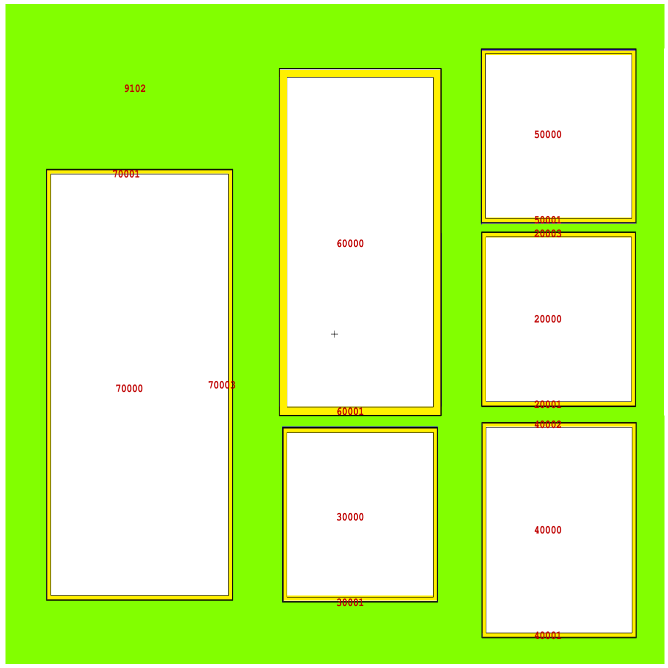

Figure 19. The entrances to the HFIR CGH neutron guides near the exit of the shutter looking downstream. The beamlines are NB1 (50000), NB2 (60000), NB3 (20000), NB4 (40000), NB5 (30000), and NB6 (70000). The yellow region are the glass guide substrates in the model, and which are surrounded by a thin vacuum region and a thin steel casing.

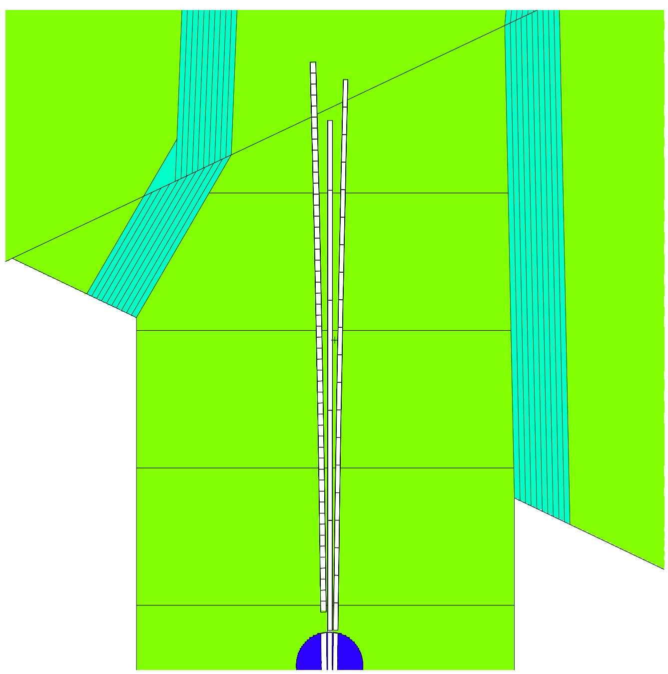

Figure 20. Three truncated beamlines from the complete model in the x-z plane. Each transverse line in the guides contains a thin stitching region.

9 REFERENCES

[1] US Product Data Association et al. Initial graphics exchange specification. iges 5.3. *ANSI. Retrieved July*, 12:2008, 1996.

[2] Joe W Durkee Jr. Mcnp geometry transformation and plotter equations. *Progress in Nuclear Energy*, 61:26–40, 2012.

[3] Matthew Frost. Specification of neutron supermirror reflectivity for cad surfaces in iges file format. unpublished, 2020.

[4] K B Grammer, F X Gallmeier, and E B Iverson. A chopper extension to model neutron transport with non-static surfaces and high-speed moving media in mcnpx 2.7. *Nucl. Instrum. Methods Phys. Res., Sect. A*, 2019.

[5] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[6] Ryan Holmes, Apr 2002.

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[8] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[9] CJ Werner. Mcnp users manual-code version 6.2 (report, la-ur-17-29981). *New Mexico: Los Alamos National Laboratory*, 2017.

[10] Peter Kjær Willendrup and Kim Lefmann. Mcstas (i): Introduction, use, and basic principles for ray-tracing simulations. *Journal of Neutron Research*, 22(1):1–16, 2020.

APPENDIX A. EXAMPLE RECTANGULAR GUIDE IGES FILE

An example IGES file for a rectangular guide segment is included here, with a discussion of each section of the file. This file contains 171 lines of text. In the case of a segmented guide with hundreds of surfaces, this can be several thousand lines long.

The header (S in column 72) and global (G in column 72) sections contain various information about the creation of the file, its name, etc. The global section also contains the units of the file.

```
CAD Geometry for "NB3_guide0179_20201007.igs" in McStas Simulation "2020S      1
-07-26_NB3_BioSANS.instr".                                             S      2
1H,,1H;,12HNB3_GUIDE1_1,16HNB3_Guide1_1.igs,11HMcStas v2.5,4Hv0.1,32,38,G      1
7,38,15,12HNB3_Guide1_1,1.,2,2HMM,32768,0.5,15H20200727.101012,0.001,999G      2
999.0,3H3xf,8HORNL_NTD,11,0,15H20200727.101012;                        G      3
```

The data (D in column 72) section contains data about each entity in the file, including the length of each line, the type, and the ID number for each entity. Each entity extends across 2 lines in the data region.

```
     116       1       1       1       0       0       0       000000000D      1
     116       0       0       1       0                   POINT       1D      2
       0       2       1       1       0       0       0       000000000D      3
       0       0       0       1       0                 NEUT_SM       1D      4
       0       3       1       1       0       0       0       000000000D      5
       0       0       0       1       0                 NEUT_SM       1D      6
     110       4       1       1       0       0       0       000000000D      7
     110       0       0       1       0                    LINE       1D      8
     110       5       1       1       0       0       0       000000000D      9
     110       0       0       1       0                    LINE       1D     10
     110       6       1       1       0       0       0       000000000D     11
     110       0       0       1       0                    LINE       1D     12
     110       7       1       1       0       0       0       000000000D     13
     110       0       0       1       0                    LINE       1D     14
     108       8       1       1       0       0       0       000000000D     15
     108       0       0       1       0                   PLANE       1D     16
     141       9       1       1       0       0       0       000000000D     17
     141       0       0       1       0                BOUNDARY       1D     18
     143      10       1       1       0       0       0       000000000D     19
     143       0       0       1       0                 BND_SRF       1D     20
     110      11       1       1       0       0       0       000000000D     21
     110       0       0       1       0                    LINE       1D     22
     110      12       1       1       0       0       0       000000000D     23
     110       0       0       1       0                    LINE       1D     24
     110      13       1       1       0       0       0       000000000D     25
     110       0       0       1       0                    LINE       1D     26
     110      14       1       1       0       0       0       000000000D     27
     110       0       0       1       0                    LINE       1D     28
     108      15       1       1       0       0       0       000000000D     29
     108       0       0       1       0                   PLANE       1D     30
```

```
     141      16       1       1       0       0       0 000000000D      31
     141       0       0       1       0          BOUNDARY       1D      32
     143      17       1       1       0       0       0 000000000D      33
     143       0       0       1       0           BND_SRF       1D      34
     110      18       1       1       0       0       0 000000000D      35
     110       0       0       1       0              LINE       1D      36
     110      19       1       1       0       0       0 000000000D      37
     110       0       0       1       0              LINE       1D      38
     110      20       1       1       0       0       0 000000000D      39
     110       0       0       1       0              LINE       1D      40
     110      21       1       1       0       0       0 000000000D      41
     110       0       0       1       0              LINE       1D      42
     108      22       1       1       0       0       0 000000000D      43
     108       0       0       1       0             PLANE       1D      44
     141      23       1       1       0       0       0 000000000D      45
     141       0       0       1       0          BOUNDARY       1D      46
     143      24       1       1       0       0       0 000000000D      47
     143       0       0       1       0           BND_SRF       1D      48
     110      25       1       1       0       0       0 000000000D      49
     110       0       0       1       0              LINE       1D      50
     110      26       1       1       0       0       0 000000000D      51
     110       0       0       1       0              LINE       1D      52
     110      27       1       1       0       0       0 000000000D      53
     110       0       0       1       0              LINE       1D      54
     110      28       1       1       0       0       0 000000000D      55
     110       0       0       1       0              LINE       1D      56
     108      29       1       1       0       0       0 000000000D      57
     108       0       0       1       0             PLANE       1D      58
     141      30       1       1       0       0       0 000000000D      59
     141       0       0       1       0          BOUNDARY       1D      60
     143      31       1       1       0       0       0 000000000D      61
     143       0       0       1       0           BND_SRF       1D      62
     110      32       1       1       0       0       0 000000000D      63
     110       0       0       1       0              LINE       1D      64
     110      33       1       1       0       0       0 000000000D      65
     110       0       0       1       0              LINE       1D      66
     110      34       1       1       0       0       0 000000000D      67
     110       0       0       1       0              LINE       1D      68
     110      35       1       1       0       0       0 000000000D      69
     110       0       0       1       0              LINE       1D      70
     108      36       1       1       0       0       0 000000000D      71
     108       0       0       1       0             PLANE       1D      72
     141      37       1       1       0       0       0 000000000D      73
     141       0       0       1       0          BOUNDARY       1D      74
       0      38       1       1       0       0       0 000000000D      75
```

```
       0       0       0       1       0                BND_SRF       1D     76
     110      39       1       1       0       0       0      000000000D     77
     110       0       0       1       0                   LINE       1D     78
     110      40       1       1       0       0       0      000000000D     79
     110       0       0       1       0                   LINE       1D     80
     110      41       1       1       0       0       0      000000000D     81
     110       0       0       1       0                   LINE       1D     82
     110      42       1       1       0       0       0      000000000D     83
     110       0       0       1       0                   LINE       1D     84
     108      43       1       1       0       0       0      000000000D     85
     108       0       0       1       0                  PLANE       1D     86
     141      44       1       1       0       0       0      000000000D     87
     141       0       0       1       0               BOUNDARY       1D     88
       0      45       1       1       0       0       0      000000000D     89
       0       0       0       1       0                BND_SRF       1D     90
```

The parameter (P in column 72) region contains the appropriate information for each entity, such as point coordinates, surface definitions, boundary definitions, or supermirror parameters. The length specified in the data section corresponds to the number of floating point numbers that appear in the parameter section, for instance. The parameters for a complete guide surface is specified by lines 7P through 19P plus line 3P for the supermirror properties.

```
116,-53.6946,40.6148,4772.99,0;                                       1P      1
0,0.990,0.022,8.000,3.044,0.003;                                      3P      2
0,0.990,0.022,8.000,3.044,0.003;                                      5P      3
110,-33.69594,60.616053480000005,4773.0461912,-73.69326000000001      7P      4
,60.61202652,4772.5847472;                                            7P      5
110,-73.69326000000001,60.61202652,4772.5847472,-76.452393375,62      9P      6
.793526604250005,5022.5599392125;                                     9P      7
110,-76.452393375,62.793526604250005,5022.5599392125,-36.7050566     11P      8
25000005,62.79752839575001,5023.018499187499;                        11P      9
110,-36.705056625000005,62.79752839575001,5023.018499187499,-33.     13P     10
69594,60.616053480000005,4773.0461912;                               13P     11
108,-2.5649961679663535e-12,-0.9999619230769968,0.00872653402871     15P     12
7162,-18.96159539896682,0,0,0,0,0;                                   15P     13
141,1,0,15,4,7,1,0,9,1,0,11,1,0,13,1,0;                              17P     14
143,0,15,1,17,0,3;                                                   19P     15
110,-73.69326000000001,20.613546520000003,4772.9338087999995,-33     21P     16
.69594,20.61757348,4773.3952528;                                     21P     17
110,-33.69594,20.61757348,4773.3952528,-36.705056625000005,22.79     23P     18
9048395750003,5023.367560787499;                                     23P     19
110,-36.705056625000005,22.799048395750003,5023.367560787499,-76     25P     20
.452393375,22.79504660425,5022.9090008124995;                        25P     21
110,-76.452393375,22.79504660425,5022.9090008124995,-73.69326000     27P     22
000001,20.613546520000003,4772.9338087999995;                        27P     23
108,2.5651739370245855e-12,0.9999619230769968,-0.008726534028717     29P     24
```

```
172,-21.038407679920557,0,0,0,0,0;                                    29P     25
141,1,0,29,4,21,1,0,23,1,0,25,1,0,27,1,0;                            31P     26
143,0,29,1,31,0,3;                                                    33P     27
110,-33.69594,60.616053480000005,4773.0461912,-33.69594,20.61757     35P     28
348,4773.3952528;                                                     35P     29
110,-33.69594,20.61757348,4773.3952528,-36.705056625000005,22.79     37P     30
9048395750003,5023.367560787499;                                      37P     31
110,-36.705056625000005,22.799048395750003,5023.367560787499,-36     39P     32
.705056625000005,62.79752839575001,5023.018499187499;                39P     33
110,-36.705056625000005,62.79752839575001,5023.018499187499,-33.     41P     34
69594,60.616053480000005,4773.0461912;                                41P     35
108,-0.9999275590766012,-0.00010503672485920939,-0.0120360112328     43P     36
24318,-23.76130544881235,0,0,0,0,0;                                   43P     37
141,1,0,43,4,35,1,0,37,1,0,39,1,0,41,1,0;                            45P     38
143,0,43,1,45,0,5;                                                    47P     39
110,-73.69326000000001,60.61202652,4772.5847472,-73.693260000000     49P     40
01,20.613546520000003,4772.9338087999995;                            49P     41
110,-73.69326000000001,20.613546520000003,4772.9338087999995,-76     51P     42
.452393375,22.79504660425,5022.9090008124995;                        51P     43
110,-76.452393375,22.79504660425,5022.9090008124995,-76.45239337     53P     44
5,62.793526604250005,5022.5599392125;                                 53P     45
110,-76.452393375,62.793526604250005,5022.5599392125,-73.6932600     55P     46
0000001,60.61202652,4772.5847472;                                     55P     47
108,-0.9999390955786669,-9.631076799760267e-05,-0.01103611605383     57P     48
6068,21.012135016952456,0,0,0,0,0;                                    57P     49
141,1,0,57,4,49,1,0,51,1,0,53,1,0,55,1,0;                            59P     50
143,0,57,1,59,0,5;                                                    61P     51
110,-33.69594,60.616053480000005,4773.0461912,-73.69326000000001     63P     52
,60.61202652,4772.5847472;                                            63P     53
110,-73.69326000000001,60.61202652,4772.5847472,-73.693260000000     65P     54
01,20.613546520000003,4772.9338087999995;                            65P     55
110,-73.69326000000001,20.613546520000003,4772.9338087999995,-33     67P     56
.69594,20.61757348,4773.3952528;                                      67P     57
110,-33.69594,20.61757348,4773.3952528,-33.69594,60.616053480000     69P     58
005,4773.0461912;                                                     69P     59
108,0.011536544489114637,-0.008725958593303815,-0.99989537742099     71P     60
73,-4773.4644906814465,0,0,0,0,0;                                     71P     61
141,1,0,71,4,63,1,0,65,1,0,67,1,0,69,1,0;                            73P     62
0,0,71,1,73;                                                          75P     63
110,-36.705056625000005,62.79752839575001,5023.018499187499,-76.     77P     64
452393375,62.793526604250005,5022.5599392125;                        77P     65
110,-76.452393375,62.793526604250005,5022.5599392125,-76.4523933     79P     66
75,22.79504660425,5022.9090008124995;                                 79P     67
110,-76.452393375,22.79504660425,5022.9090008124995,-36.70505662     81P     68
5000005,22.799048395750003,5023.367560787499;                        81P     69
```

```
110,-36.705056625000005,22.799048395750003,5023.367560787499,-36        83P      70
.705056625000005,62.79752839575001,5023.018499187499;                   83P      71
108,0.011536544489098632,-0.008725958593303818,-0.99989537742099        85P      72
75,-5023.46439618901,0,0,0,0,0;                                         85P      73
141,1,0,85,4,77,1,0,79,1,0,81,1,0,83,1,0;                               87P      74
0,0,85,1,87;                                                            89P      75
S      2G      3D      90P     75                                         T0000001
```