OpenMC Workshop

CSG Briefer

ANS Student Conference

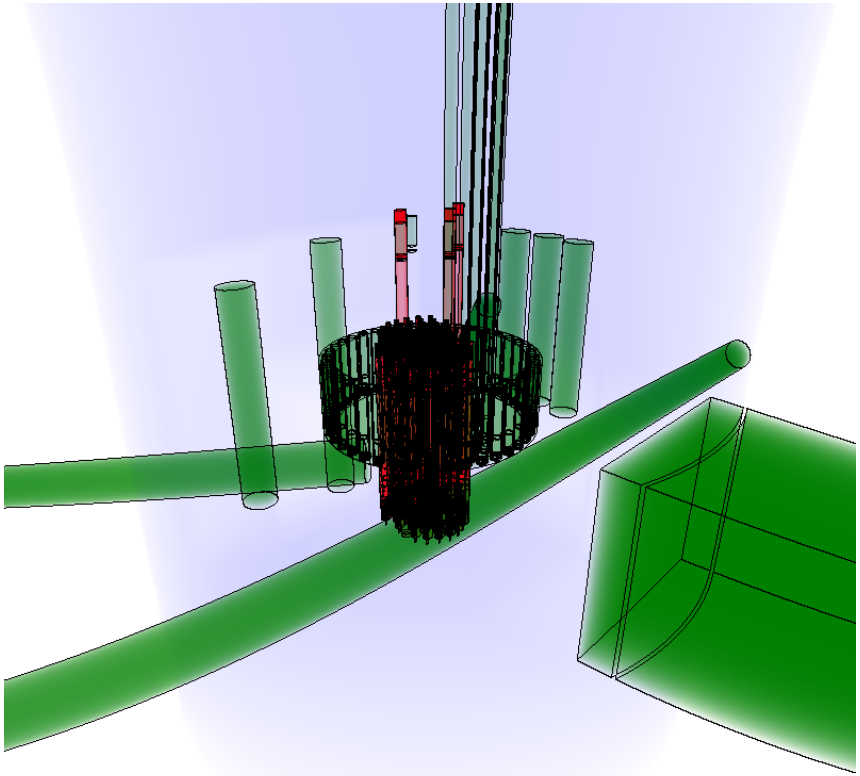
April 13, 2023

Constructive Solid Geometry (CSG)

How can we represent geometry on a computer, to get something complex like a reactor?

- Surface meshes
  (e.g. videogame characters)
- Volume meshes
  (e.g. FEM solves)
- Voxelization
  (e.g. Minecraft)
- Constructive solid geometry
  (e.g. right image)

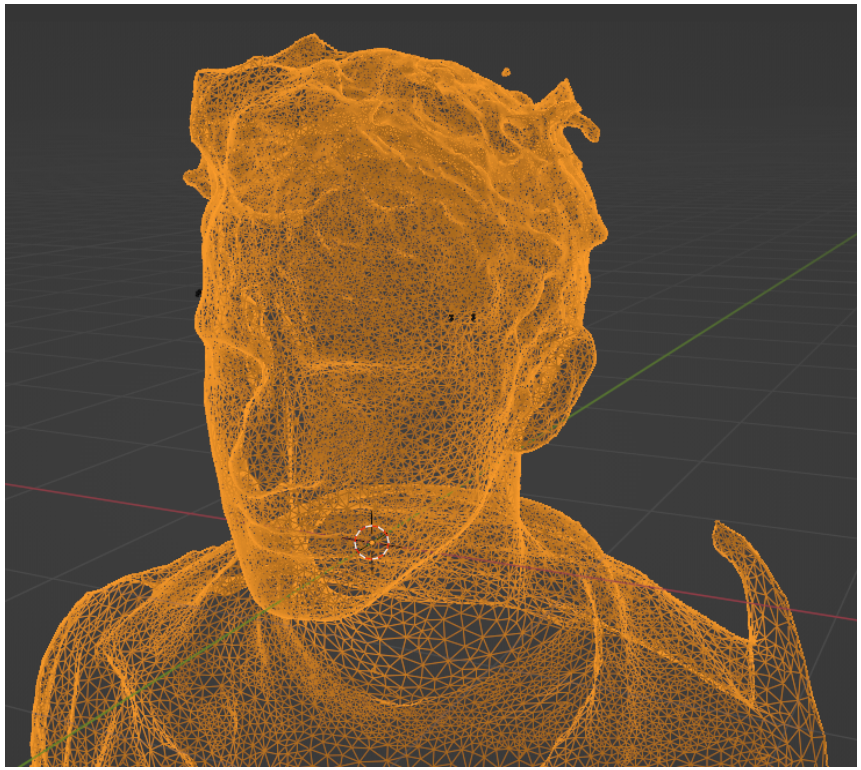
Image: own work



TRIGA reactor ex-core components.
Model courtesy JSI, Slovenia.

Surface meshes

- Method of choice in computer graphics
- OpenMC can use surface meshes via DAGMC
- We could, for example, calculate the k eigenvalue of my friend Lorenzo if he were made of plutonium

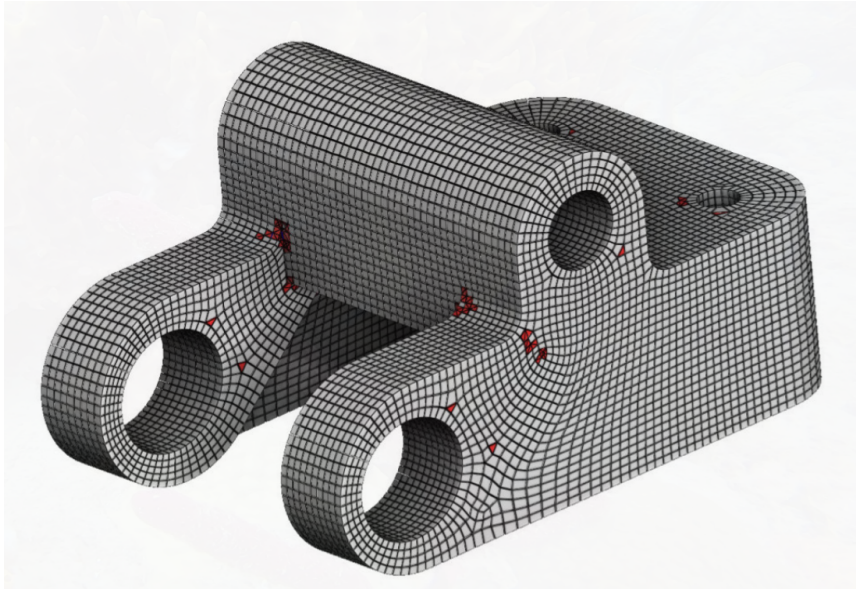


Lorenzo's head triangulation

resulting surface

Volume meshes

- Commonly used to calculate stresses in parts, heat transfer, CFD, etc.
- OpenMC **cannot** natively define problem geometries using volume meshes, but tallying **can** be done on volume meshes
- Tools associated with DAGMC can be used to extract surface meshes from volume meshes for use with OpenMC
- More info: svalinn.github.io/DAGMC/usersguide/

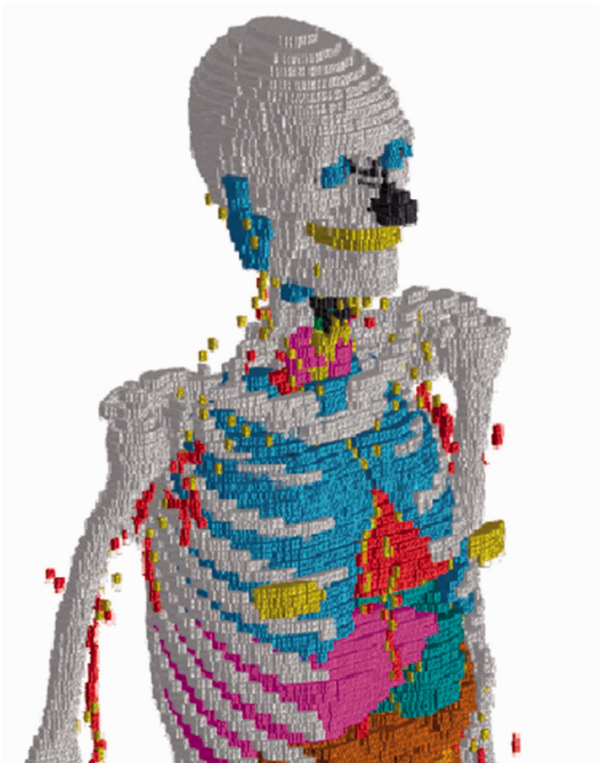


For example, hex meshes are lots of little deformed bricks put together

[0]Image: *Hexahedral Meshing: Mind the Gap!*. Ray, Sokolov, et. al. hal.inria.fr/hal-01551603/document

Voxelization

- Many radiation simulating programs employ voxelization
- Los Alamos Nat'l Lab's PARTISN
- numerous health physics codes
- **OpenMC does not rely on or use voxelization**



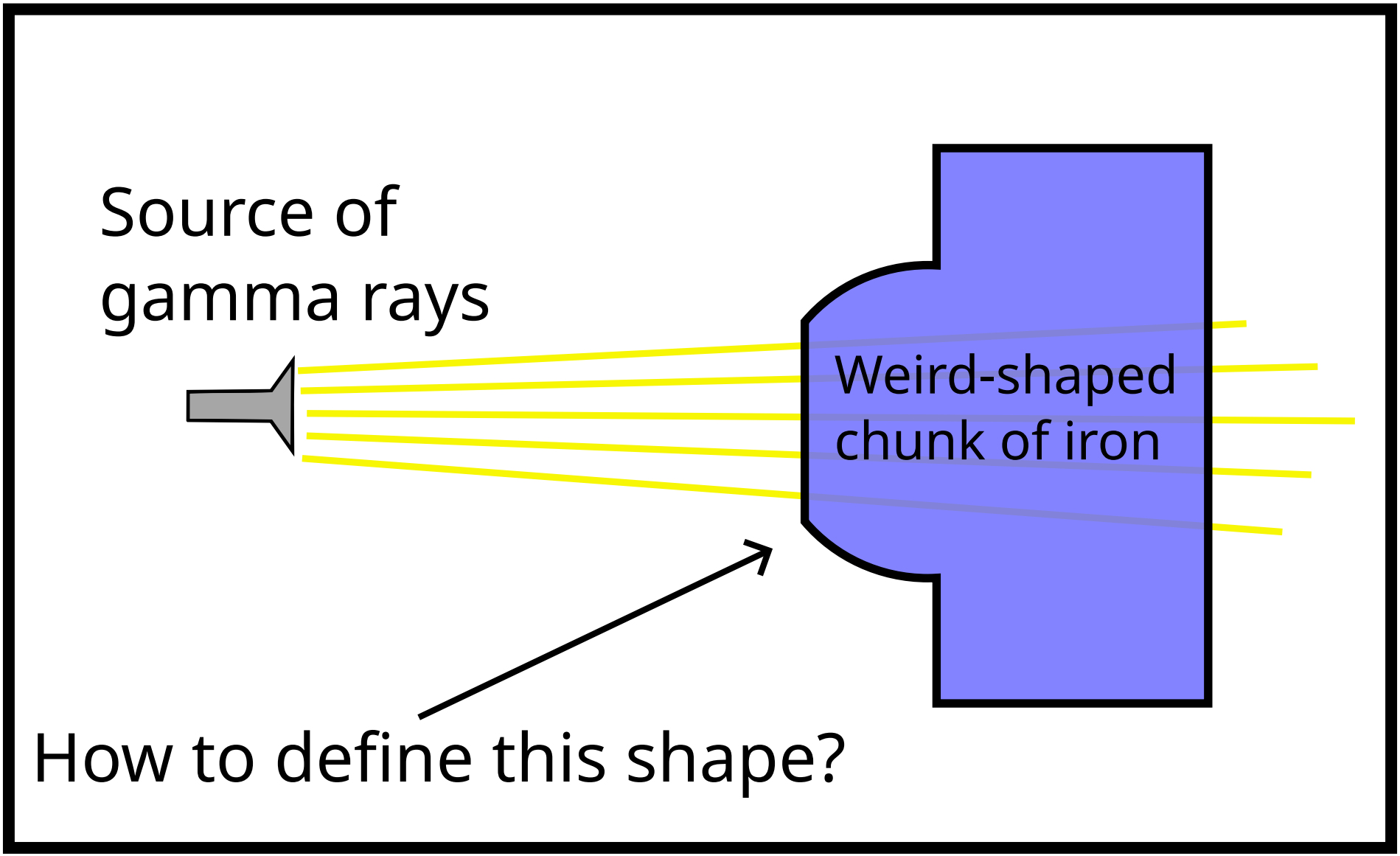
Medical phantom

[0]Image: *Computational phantoms, ICRP/ICRU, and further developments*. Zankl, Becker et. al. Annals of the ICRP, 2018.

Computational Solid Geometry

- Let's start from the absolute basics!
- Suppose we want to solve a 2D problem like this one:
- How to define using OpenMC's CSG?

Computational Solid Geometry

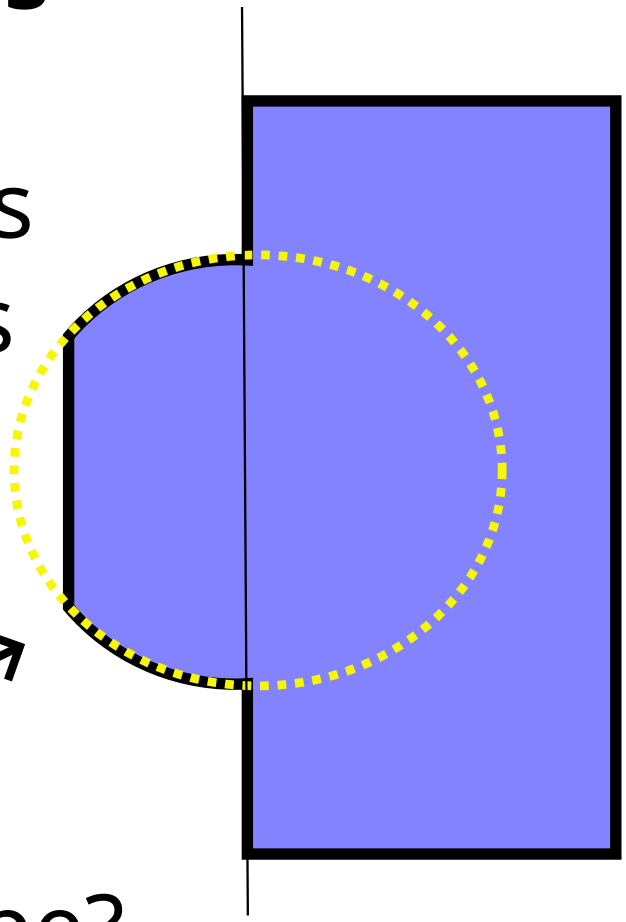


Computational Solid Geometry

Step 1) define surfaces

Step 2) define volumes

Step 3) define materials that fill volumes



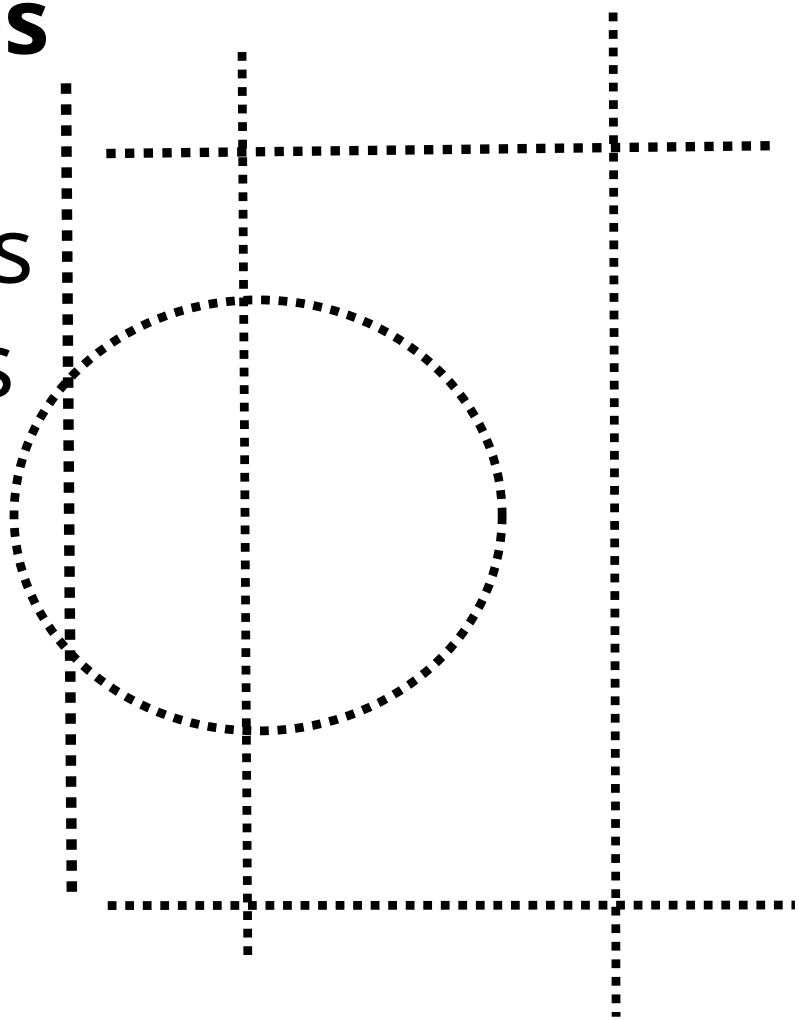
How to define this shape?

Computational Solid Geometry

Step 1) define surfaces

Step 2) define volumes

Step 3) define materials that fill volumes



Computational Solid Geometry

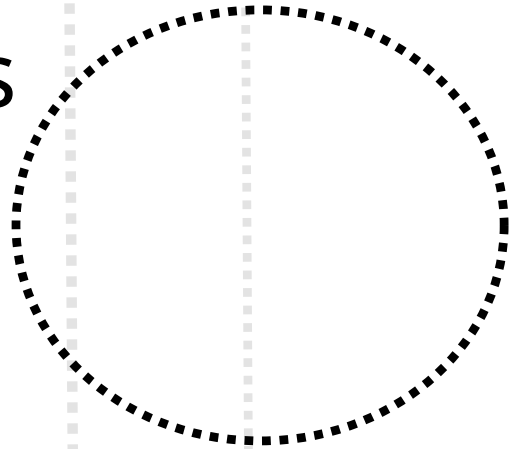
Step 1) define surfaces

Step 2) define volumes

Step 3) define materials that fill volumes

openmc.ZCylinder

$$(x - x_0)^2 + (y - y_0)^2 = r^2$$



Computational Solid Geometry

Step 1) define surfaces

Step 2) define volumes

Step 3) define materials that fill volumes

openmc.XPlane

$$x - x_0 = 0$$

$$x - x_1 = 0$$

$$x - x_2 = 0$$

Computational Solid Geometry

Step 1) define surfaces

Step 2) define volumes

Step 3) define materials that fill volumes

openmc.YPlane

$$y - y_0 = 0$$

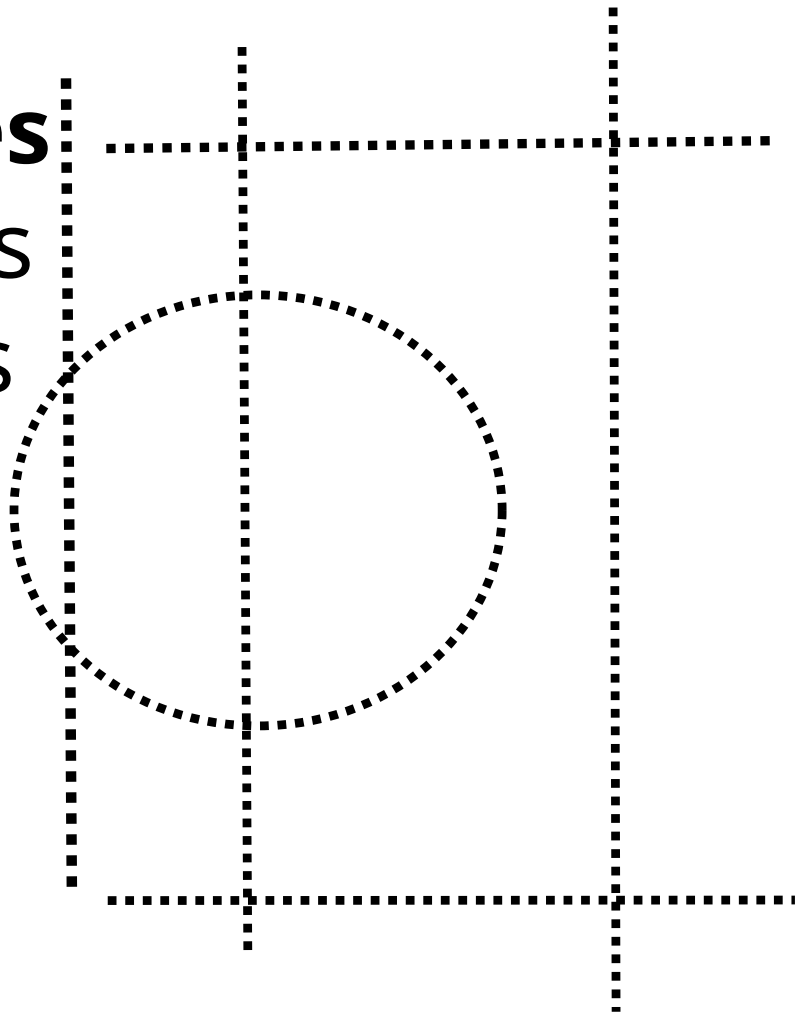
$$y - y_1 = 0$$

Computational Solid Geometry

Step 1) define surfaces

Step 2) define volumes

Step 3) define materials that fill volumes

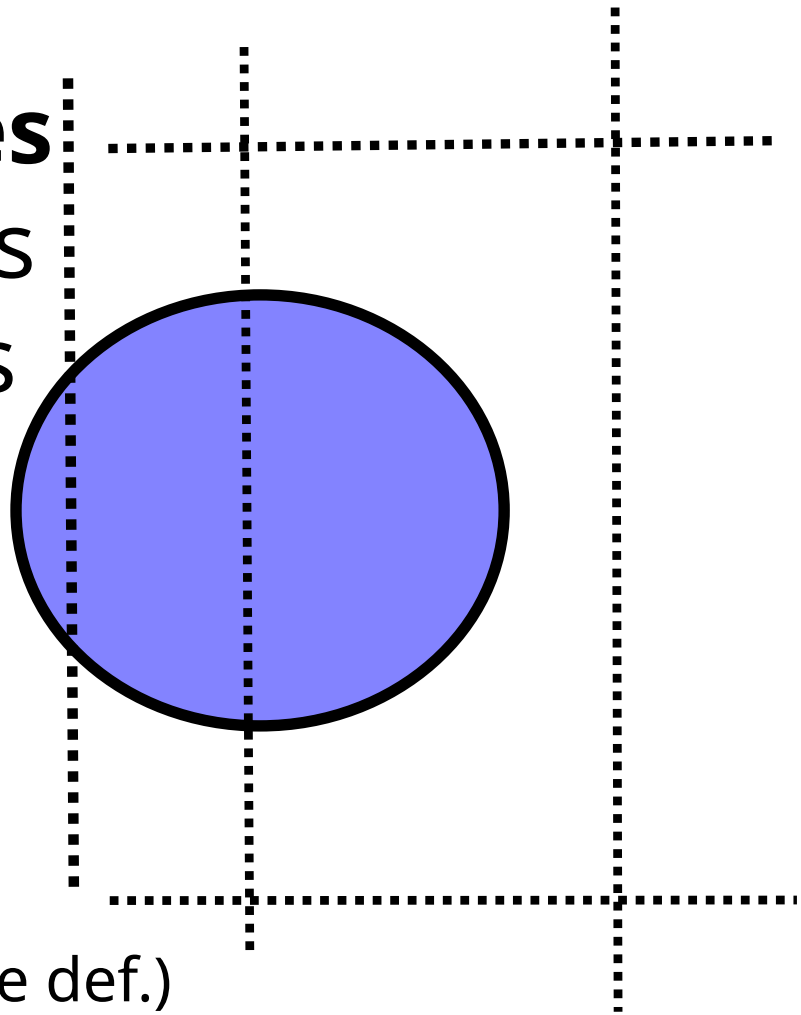


Computational Solid Geometry

Step 1) define surfaces

Step 2) define volumes

Step 3) define materials that fill volumes



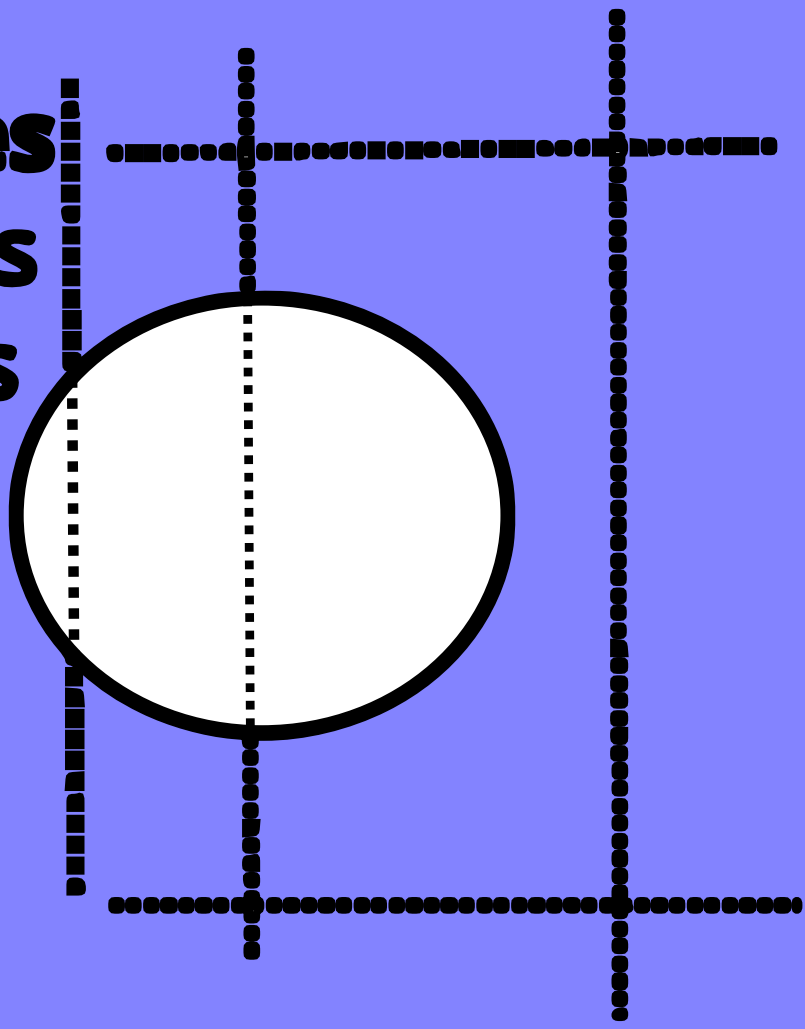
openmc.Region

$$(x - x_0)^2 + (y - y_0)^2 < r^2$$

(takes reference of original surface def.)

Computational Solid Geometry

Step 1) define surfaces
Step 2) define volumes
Step 3) define materials that fill volumes



openmc.Region

$$(x - x_0)^2 + (y - y_0)^2 > r^2$$

Computational Solid Geometry

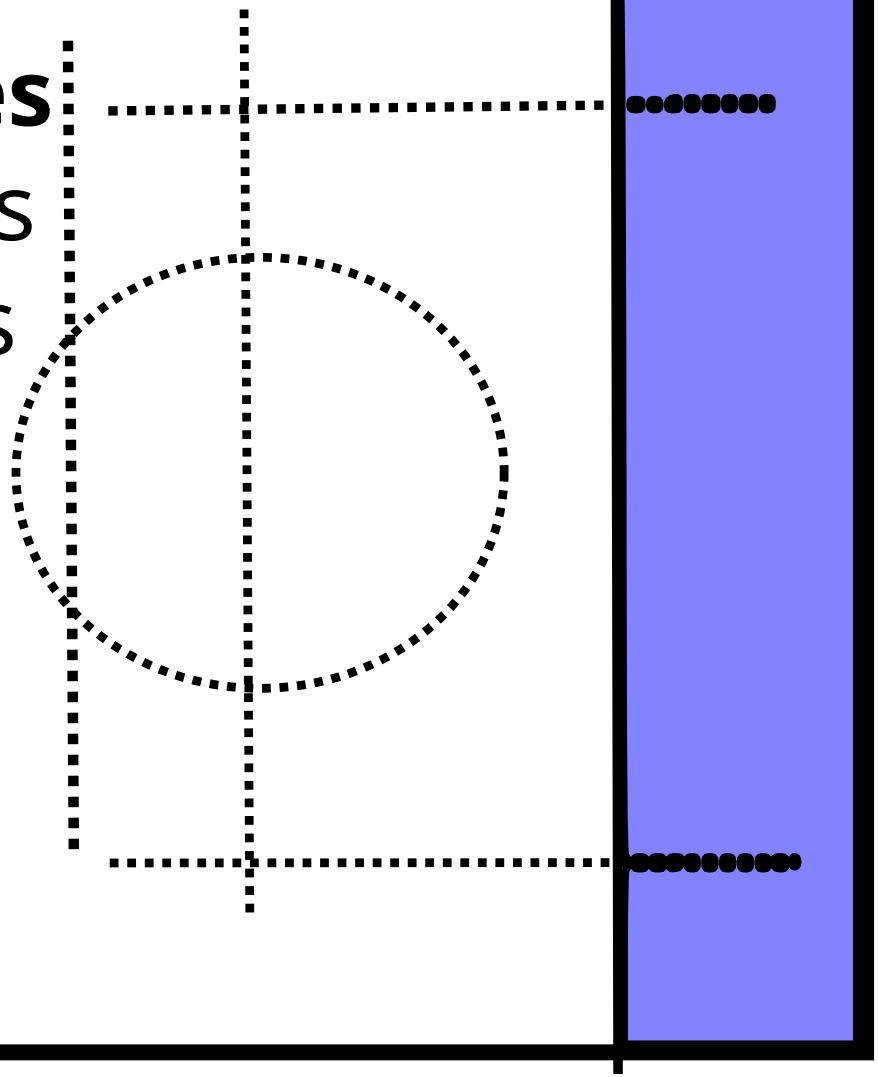
Step 1) define surfaces

Step 2) define volumes

Step 3) define materials that fill volumes

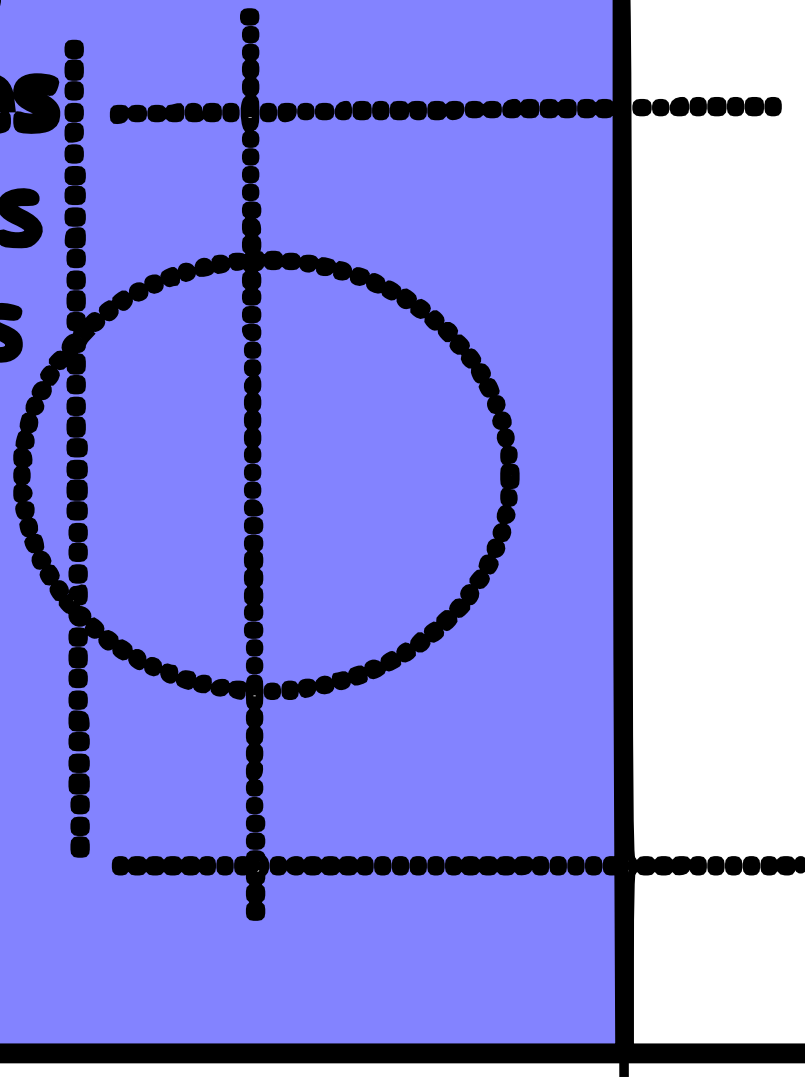
openmc.Region

$$x - x_0 > 0$$



Computational Solid Geometry

Step 1) define surfaces
Step 2) define volumes
Step 3) define materials that fill volumes



openmc.Region

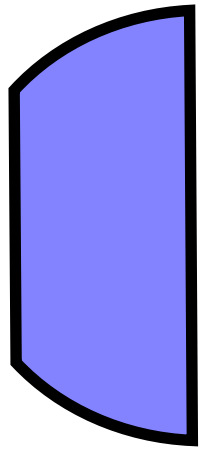
$$x - x_0 < 0$$

Computational Solid Geometry

Step 1) define surfaces

Step 2) define volumes

Step 3) define materials
that fill volumes



openmc.Region &
openmc.Region

(join with intersection
operator)

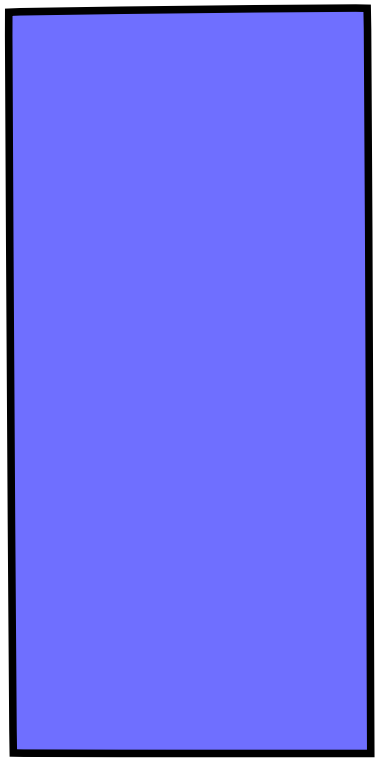
Computational Solid Geometry

Step 1) define surfaces

Step 2) define volumes

Step 3) define materials
that fill volumes

openmc.Region &
openmc.Region &
openmc.Region &
openmc.Region
(intersect all plane regions)

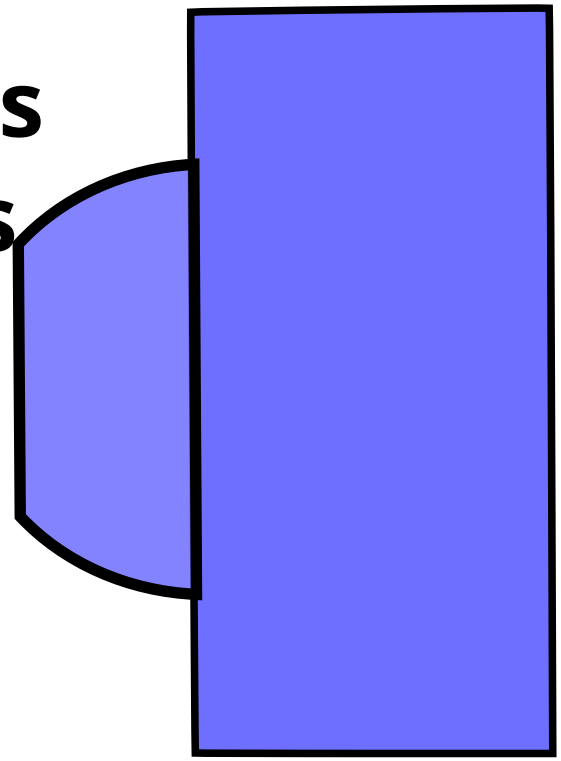


Computational Solid Geometry

Step 1) define surfaces

Step 2) define volumes

Step 3) define materials that fill volumes



region1.fill = iron

region2.fill = iron