Little Beast engineering

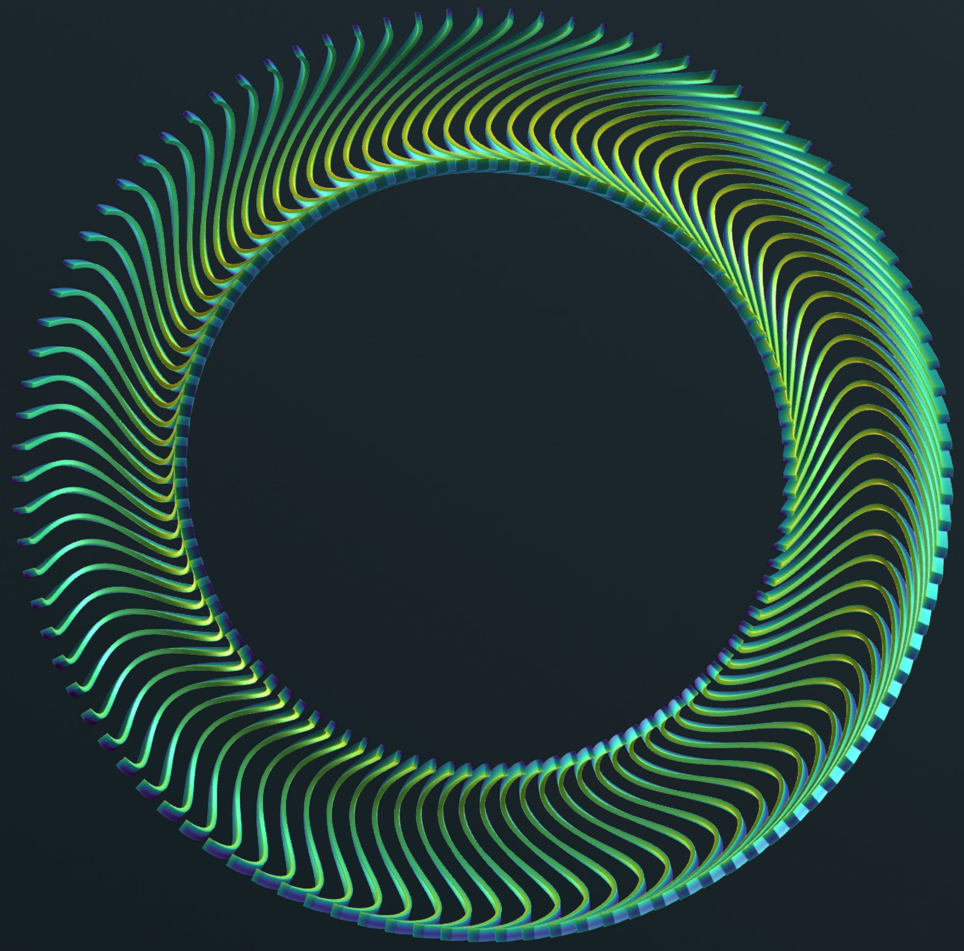
RAT | An Open-Source Magnet-Design Tool

Autumn 2024

Jeroen van Nugteren

5th PBC technology Mini workshop 2024

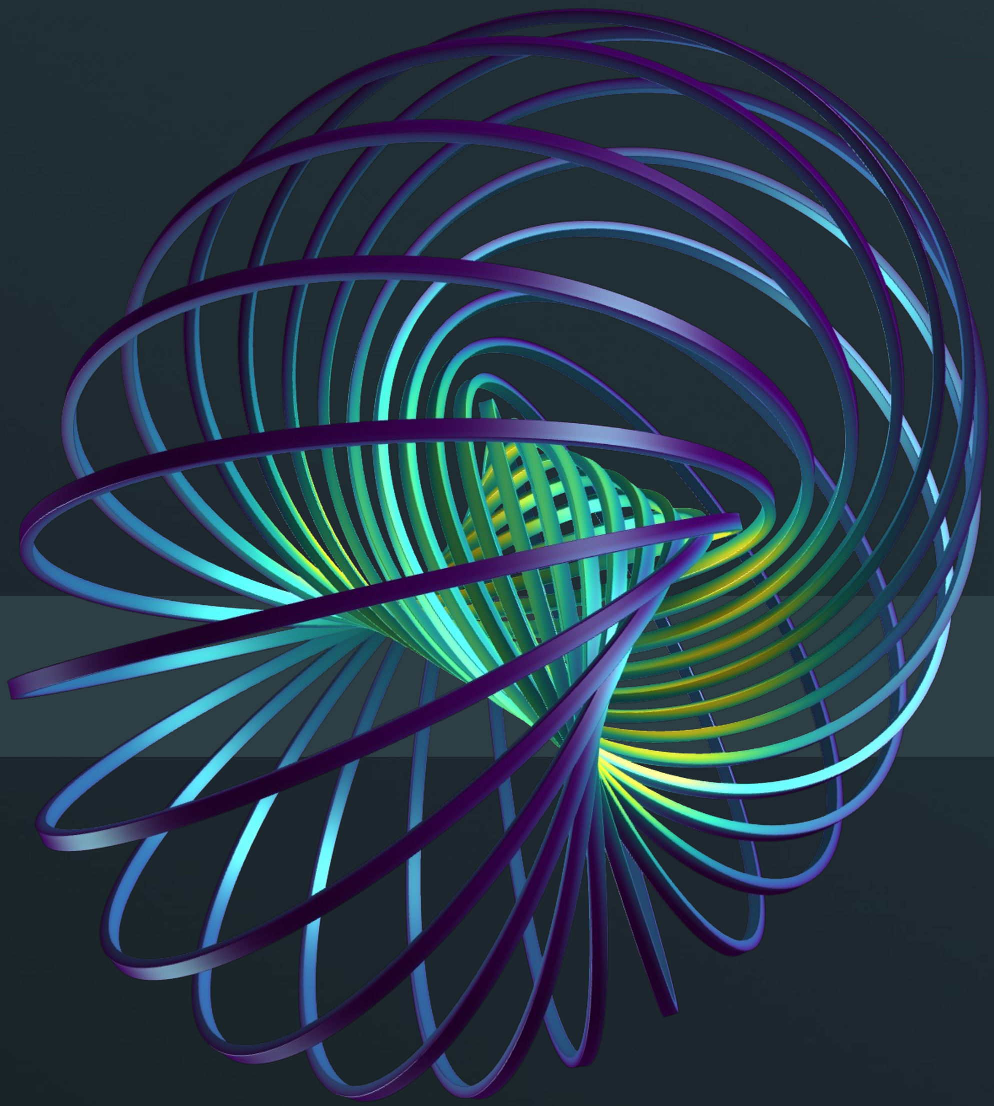
Today's topics



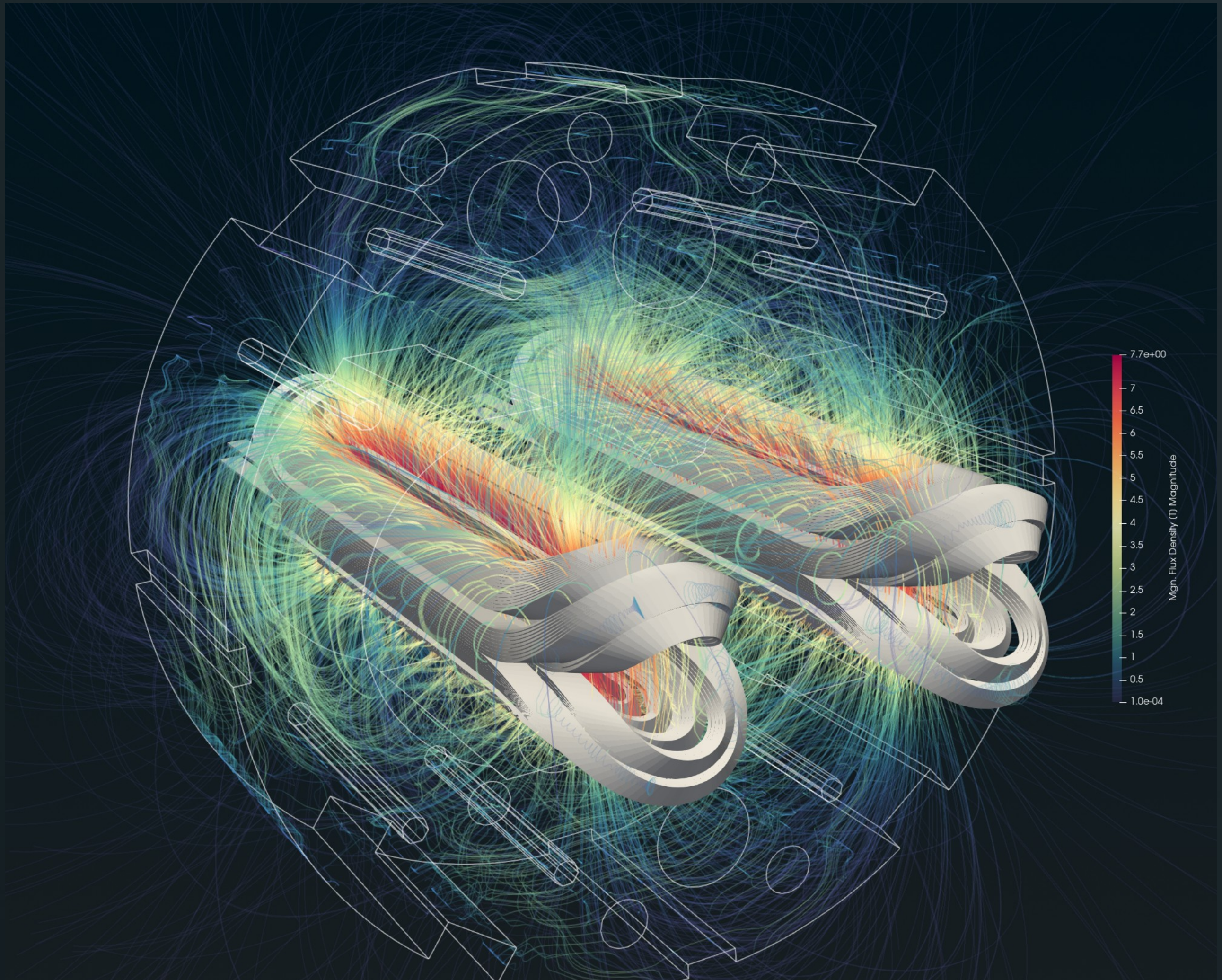
- Introduction
  - What is RAT?
  - How does RAT work?
  - How is RAT validated?
  - How can I get RAT?
- Features
  - Winding algorithm
  - Longitudinal Grading
  - Connectors
  - FreeCAD interface
  - Modeling CCT Magnets
- CMS Modeling Example
  - Building CMS magnet geometry in 10 steps
- CMS Calculations

Introduction

What is RAT and what can it do?



What is RAT ?

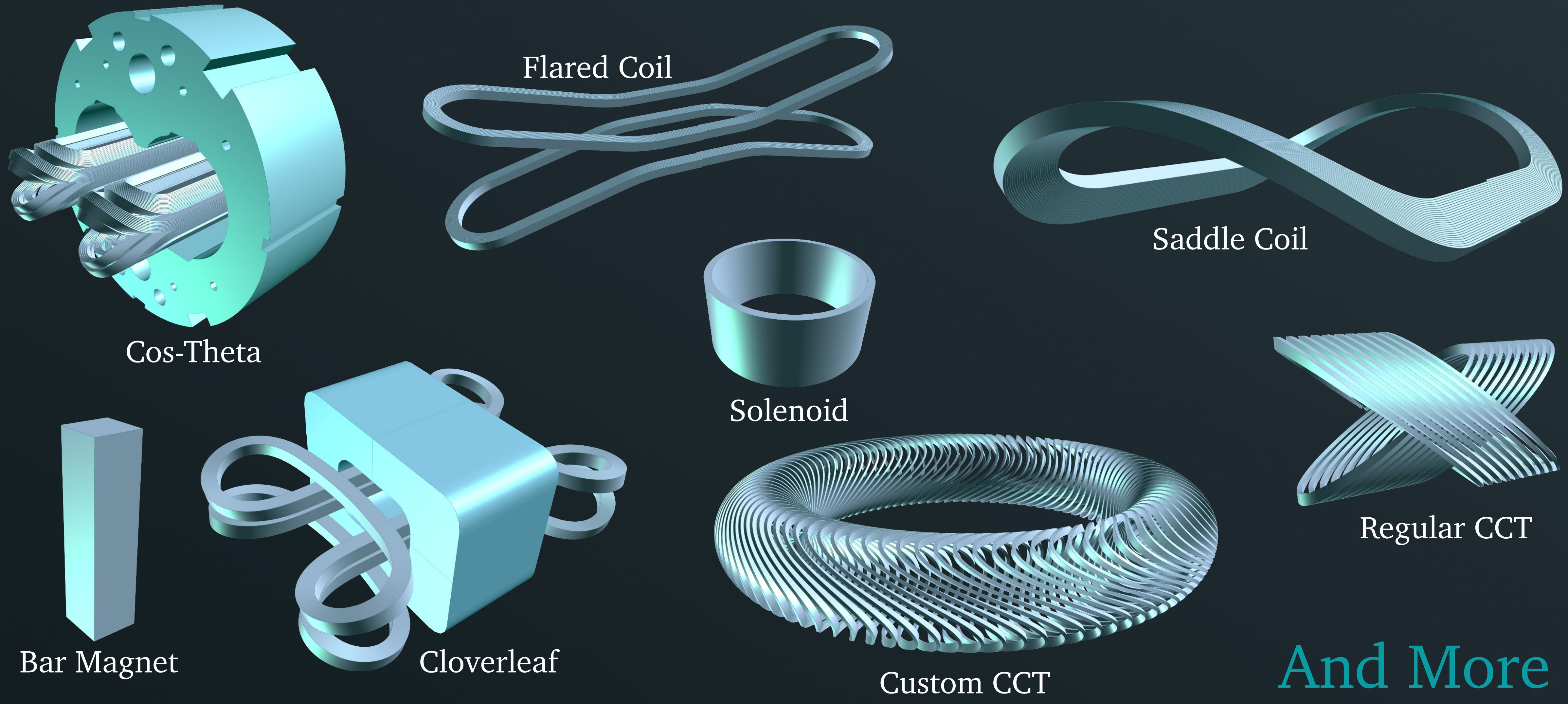


LHC Dipole in RAT

- Software for 3D magneto-statics including non-linear materials
- Extensive coil/yoke modeler with numerous templates
- Versatile calculation options
- Never ever mesh the air:
  - Biot-Savart with Fast Multipole Method (FMM) for coils
  - Volume Integral Method (VIM) [1] for HB domains (i.e. Iron)

[1] V. Le-Van, G. Meunier, O. Chadebec and J. -M. Guichon, "A Magnetic Vector Potential Volume Integral Formulation for Nonlinear Magnetostatic Problems," in IEEE Transactions on Magnetics, vol. 52, no. 3, pp. 1-4, March 2016, Art no. 7002804, doi: 10.1109/TMAG.2015.2490627.

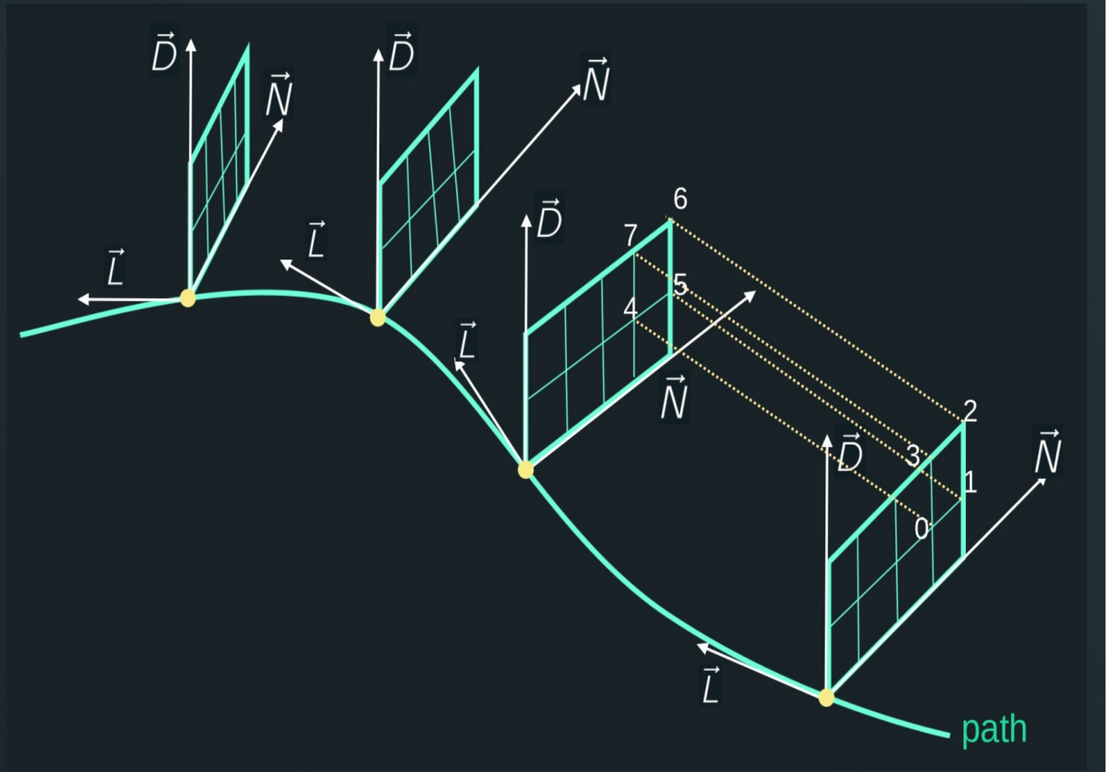
Many Built in Coil Templates



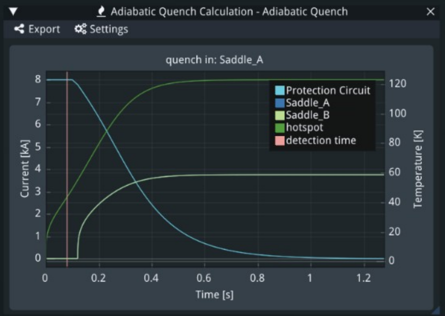
And More ...

Modeling Custom Coils in RAT

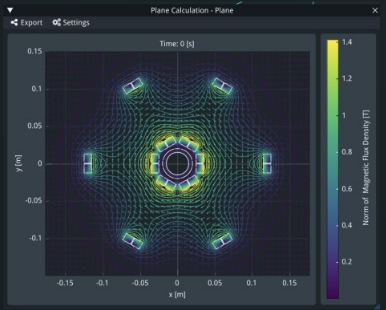
- All coil/yoke objects in RAT are constructed using:
  - A path: a 3D space curve with local coordinate system (Frame)
  - A Cross Section: a 2D area defined by a closed perimeter
- The 2D cross section is meshed:
  - Regular mesh for simple shapes
  - Distmesher for complicated shapes
- The 2D mesh is extruded along the path to create a 3D Mesh



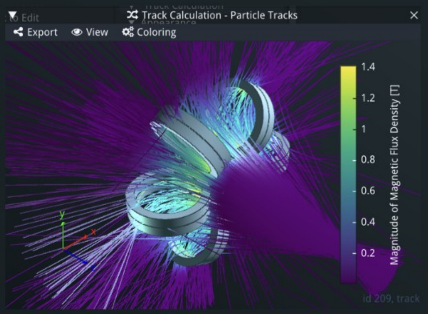
Many Built in Calculations



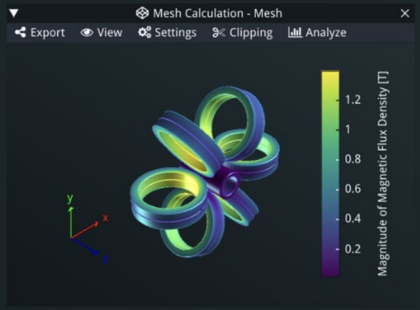
Adiabatic Quench (new)



Plane Calculation



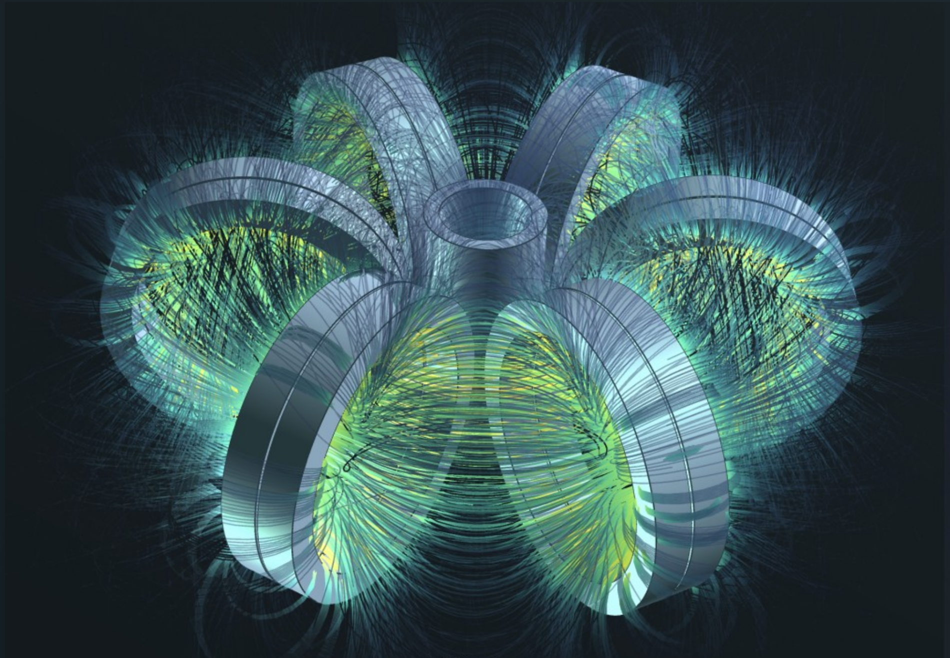
Particle Tracks



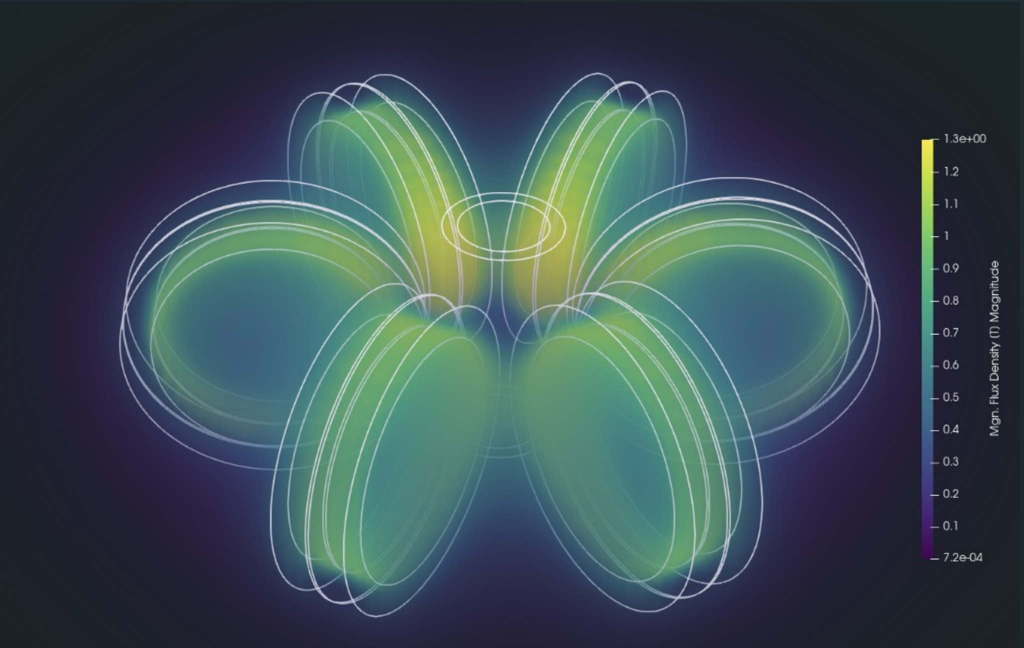
Mesh Calculation

Inductance Calculation

Line Calculation



Field Lines



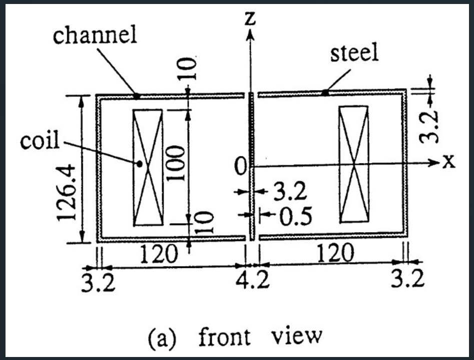
Grid (post in paraview)

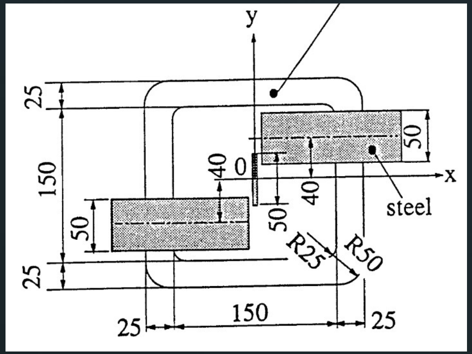
And More ...

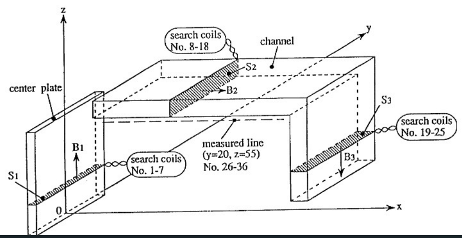
How is ЯΔ⊣ Validated?

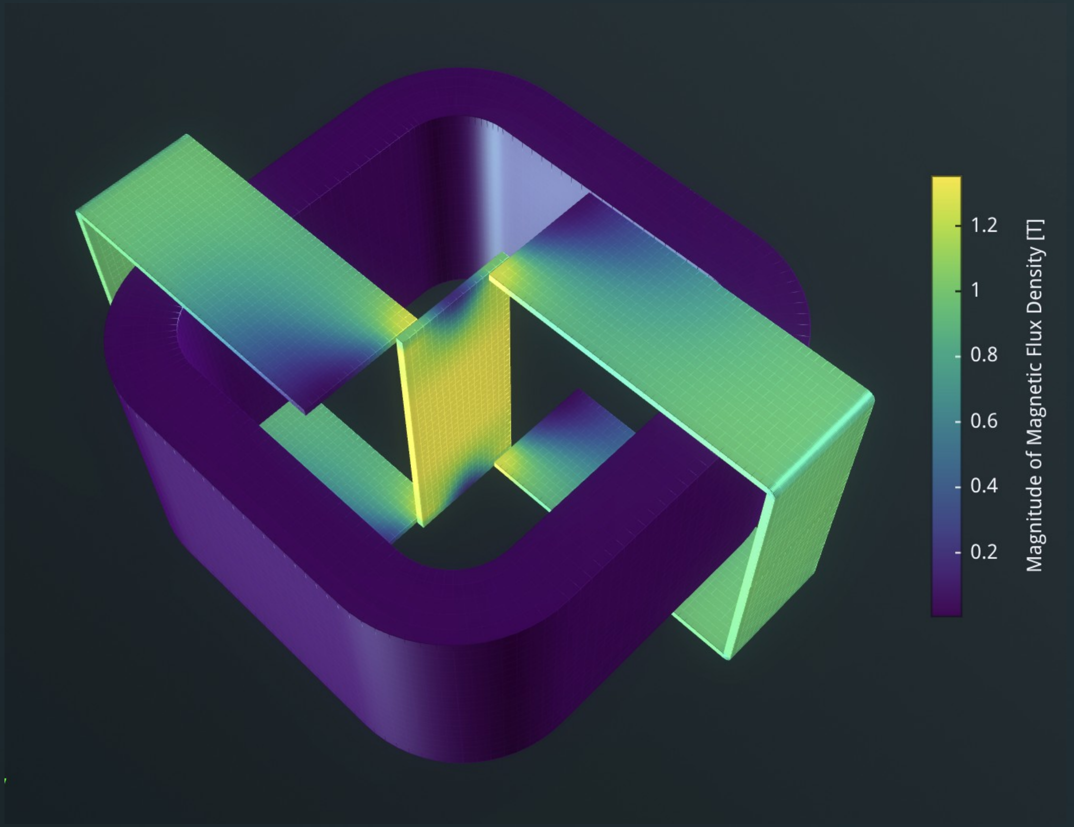
- Numerous regression tests
- Comparing to Analytical equations
  - Field on Solenoid axis
  - Field inside magnetized sphere
  - Field of iron sphere in background field
- Comparison to Soleno
  - Solenoid field everywhere
- Comparison to Pre-Calculated fields
  - Iron ring transformer
- Compumag TEAM13 problem ->

- All tests are run before updating the code on GIT (next slide)









Where can I find RAT?

Project-Rat Free

Group ID: 1475626

- **Rat-Common** — Contains common files for Rat. ★ 2
- **Rat-Models** — Coil geometry modeller and field calculation written in C++. ★ 3
- **Rat-NL** — Nonlinear magnetic materials solver written in C++. ★ 0
- **Rat-MLFMM** — Matrix based Multi-Level Fast Multipole Method for magn... ★ 4
- **Materials-CPP** — Implementation in C++ for using material data stored in js... ★ 0
- **DistMesh-CPP** — Tri- and quad-mesh generator based on distmesh written ... ★ 2
- **Rat-Docs** — Documentation for Rat. ★ 0
- **Rat-Template** — Template for setting up a new Rat project. ★ 0

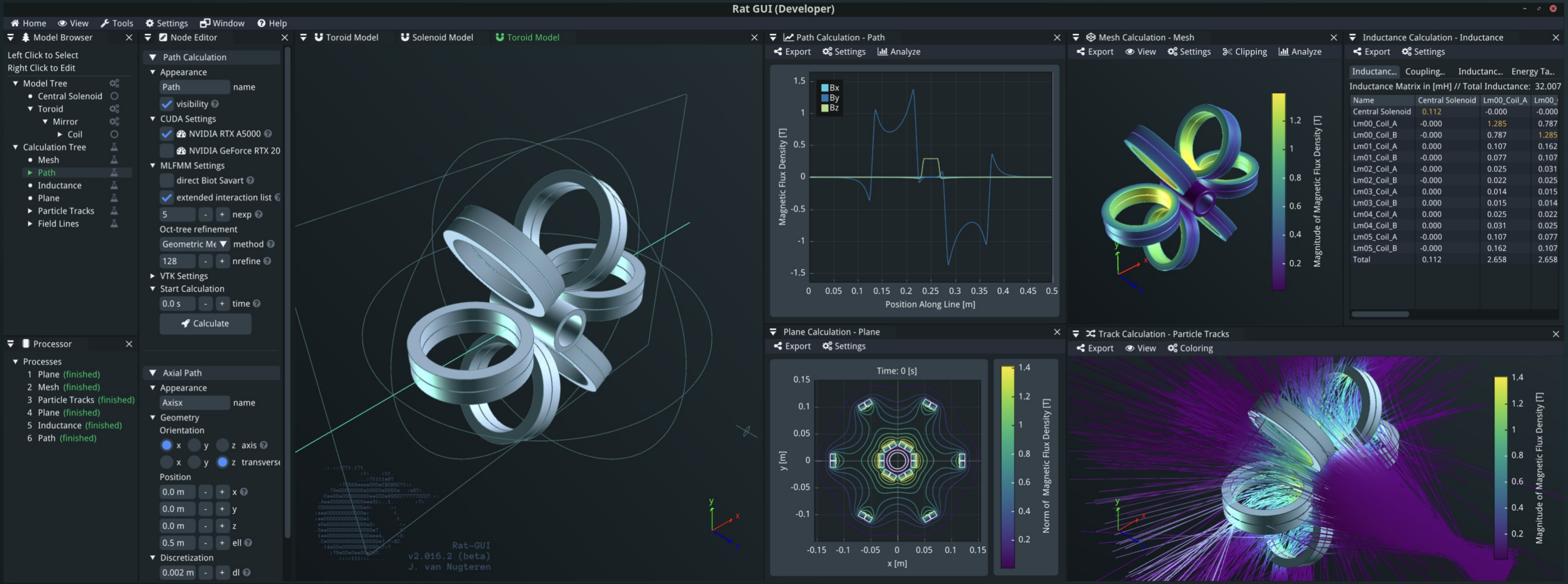
- Open Source Collection of Libraries
  - Permissive MIT license model (do what you want!)
  - Calculation Code freely available from GITLAB https://gitlab.com/Project-Rat
- Written in object oriented C++
- Output in VTK File formats (ParaView)

Windows

Graphical User Interface

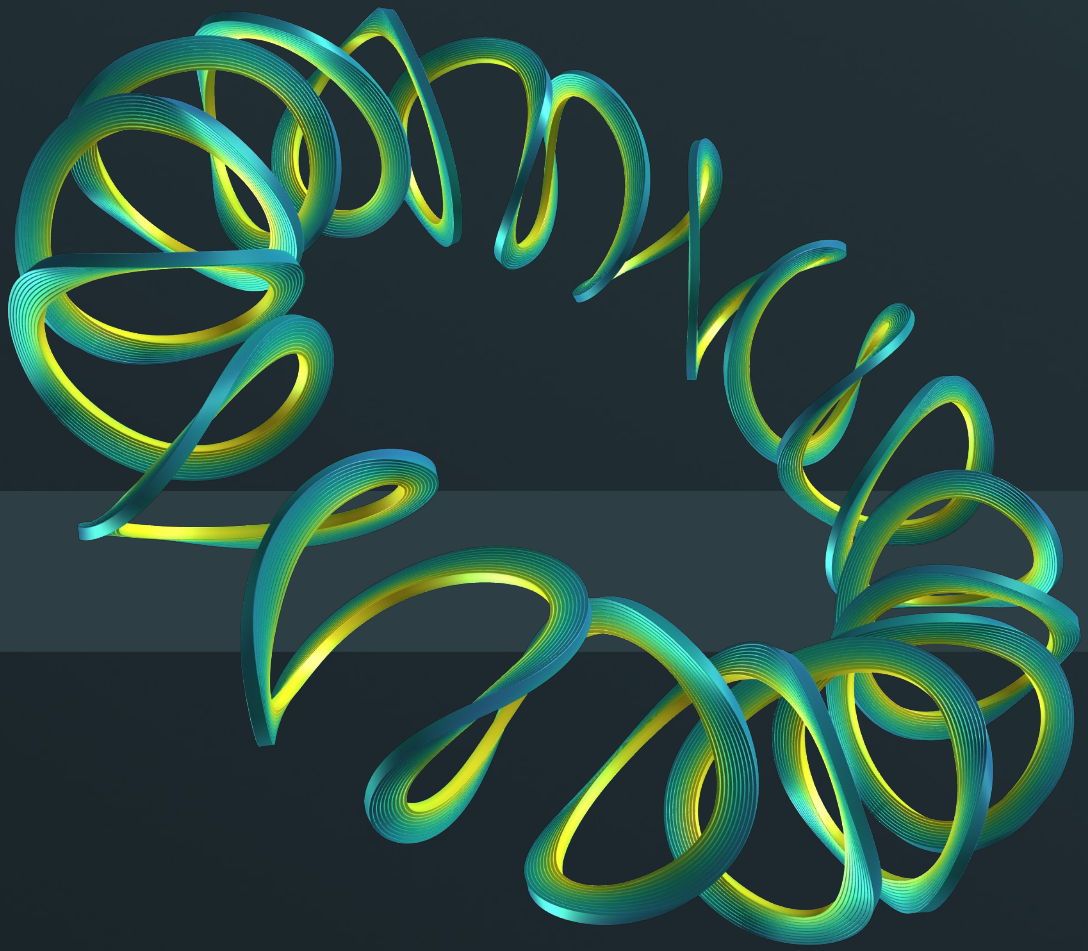
- Convenient Graphical User Interface (GUI) available for sale
  - Supports development of the open source code

Windows



Features

Useful features available in RAT

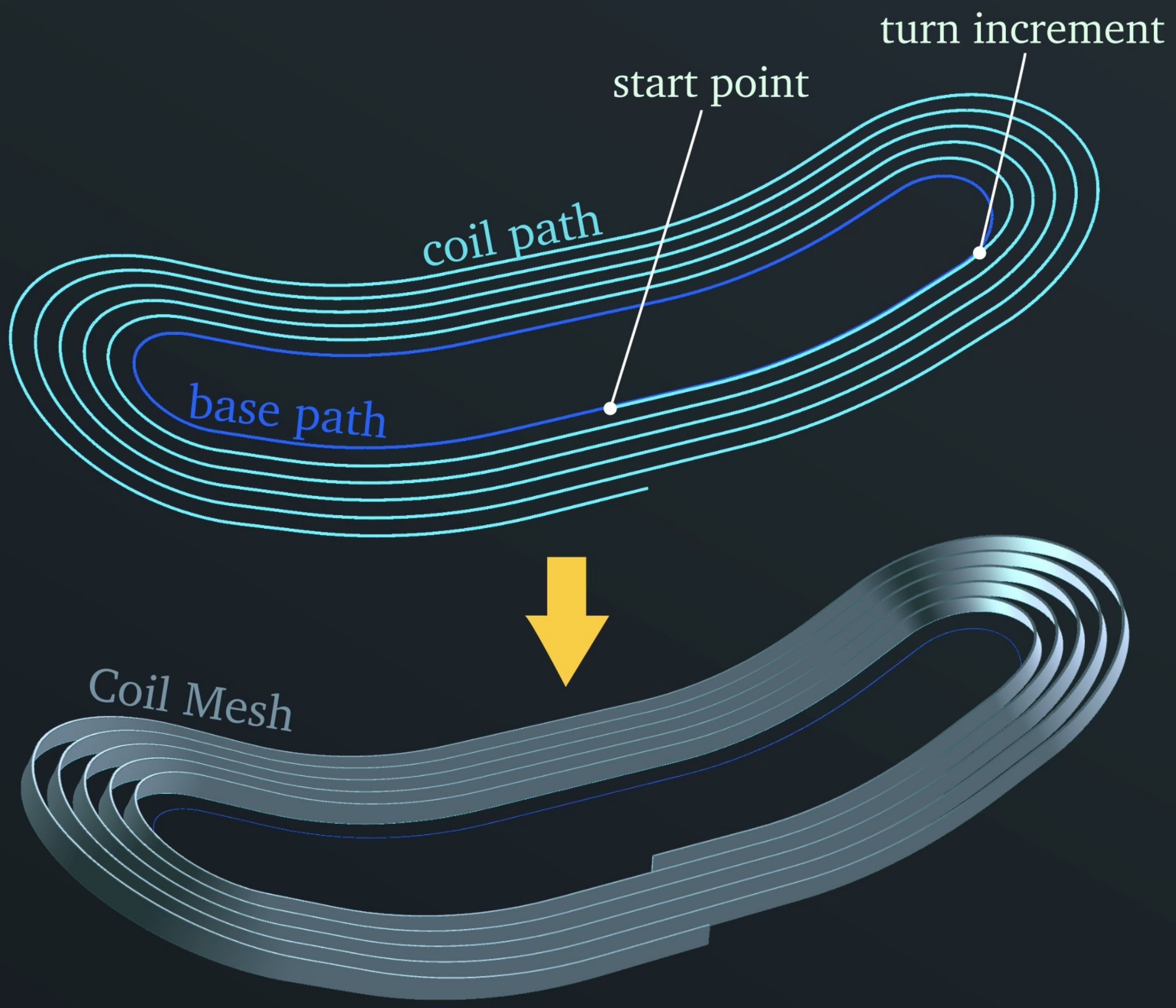


Algorithm

The coiling algorithm takes a base path and winds a coil path on top of it.

The position of the start of the winding, the increment of the turn and the end of the winding can be specified.

After creating a path for the coil it can be used together with a cross section to draw the cable.

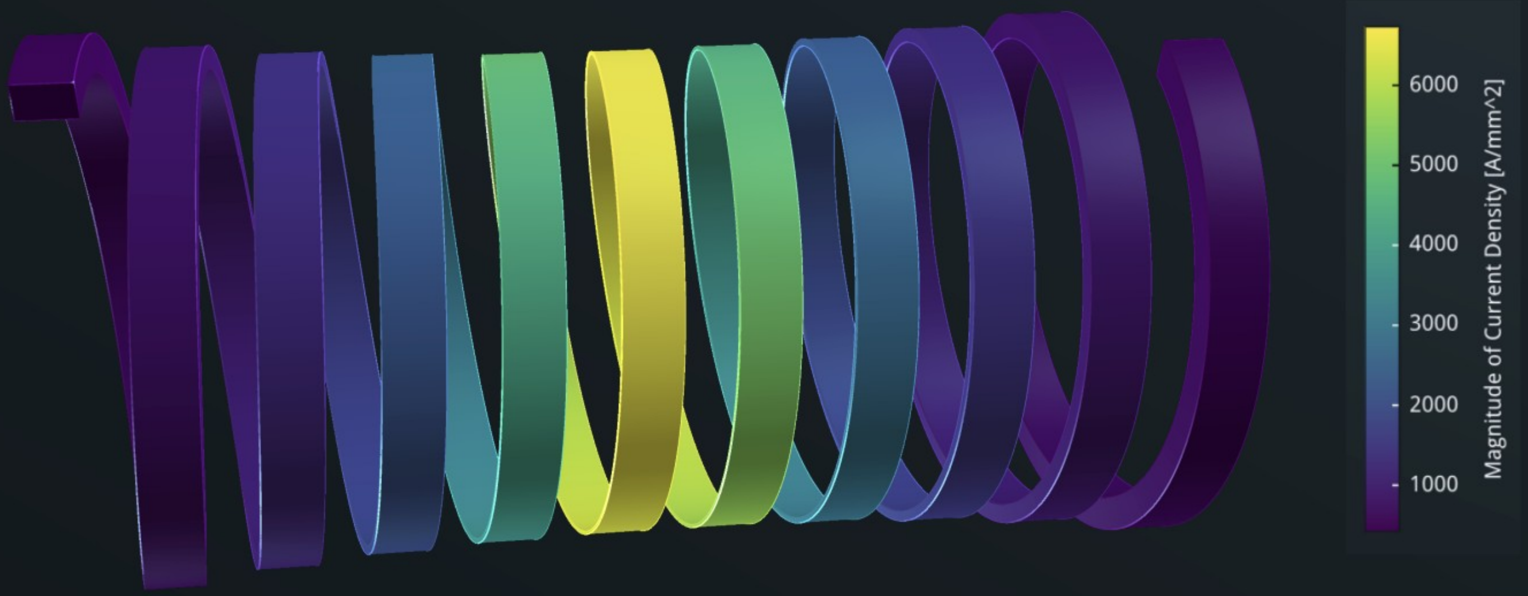


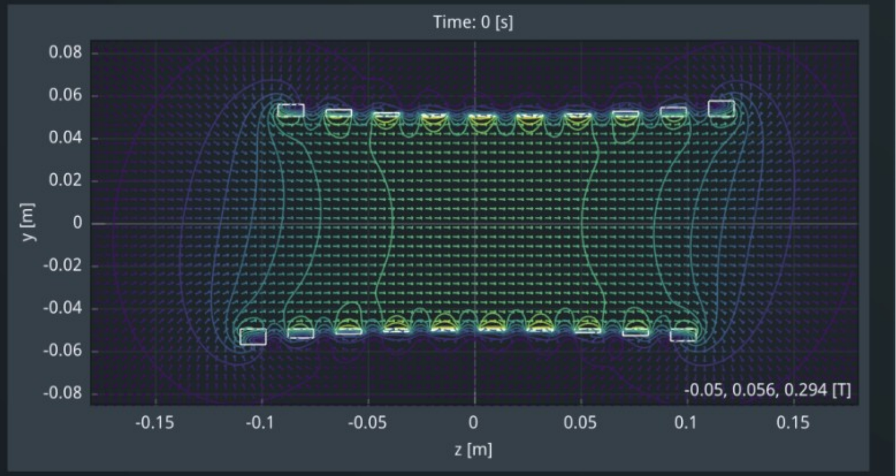
Longitudinal Grading

Especially in HTS coils it is found that grading the cable along its length by locally changing the number of tapes can save up to factor 2 of conductor.

Rat allows cables with varying thickness while correctly preserving current.

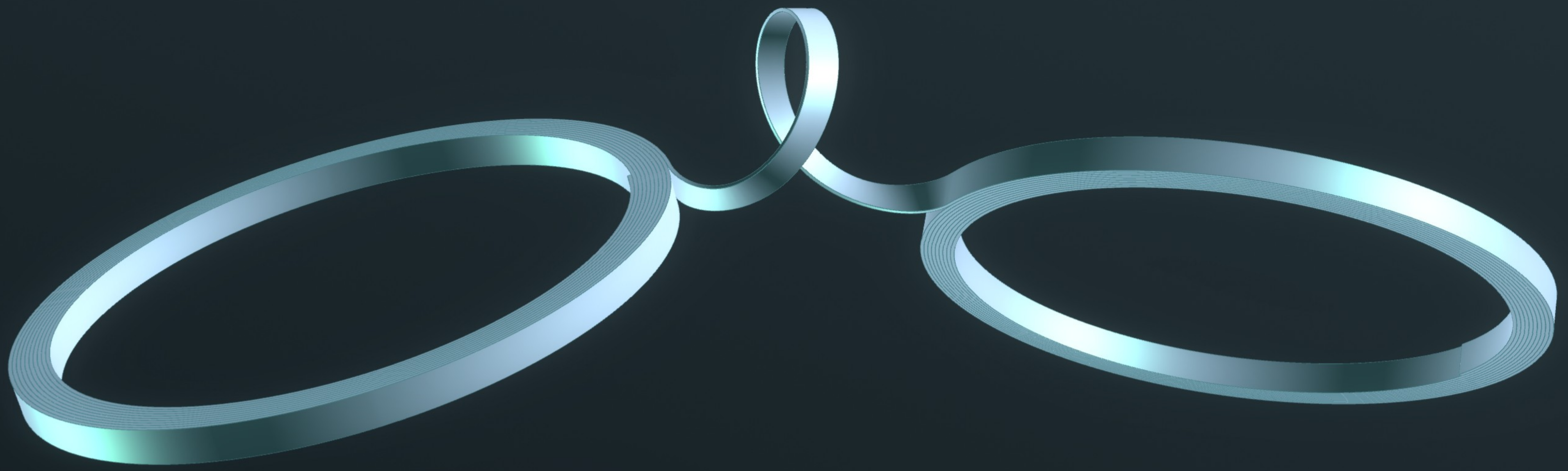
See here an example of a graded HTS solenoid.





Rat-Models
Coil geometry modeller and field calculation written in C++.
3

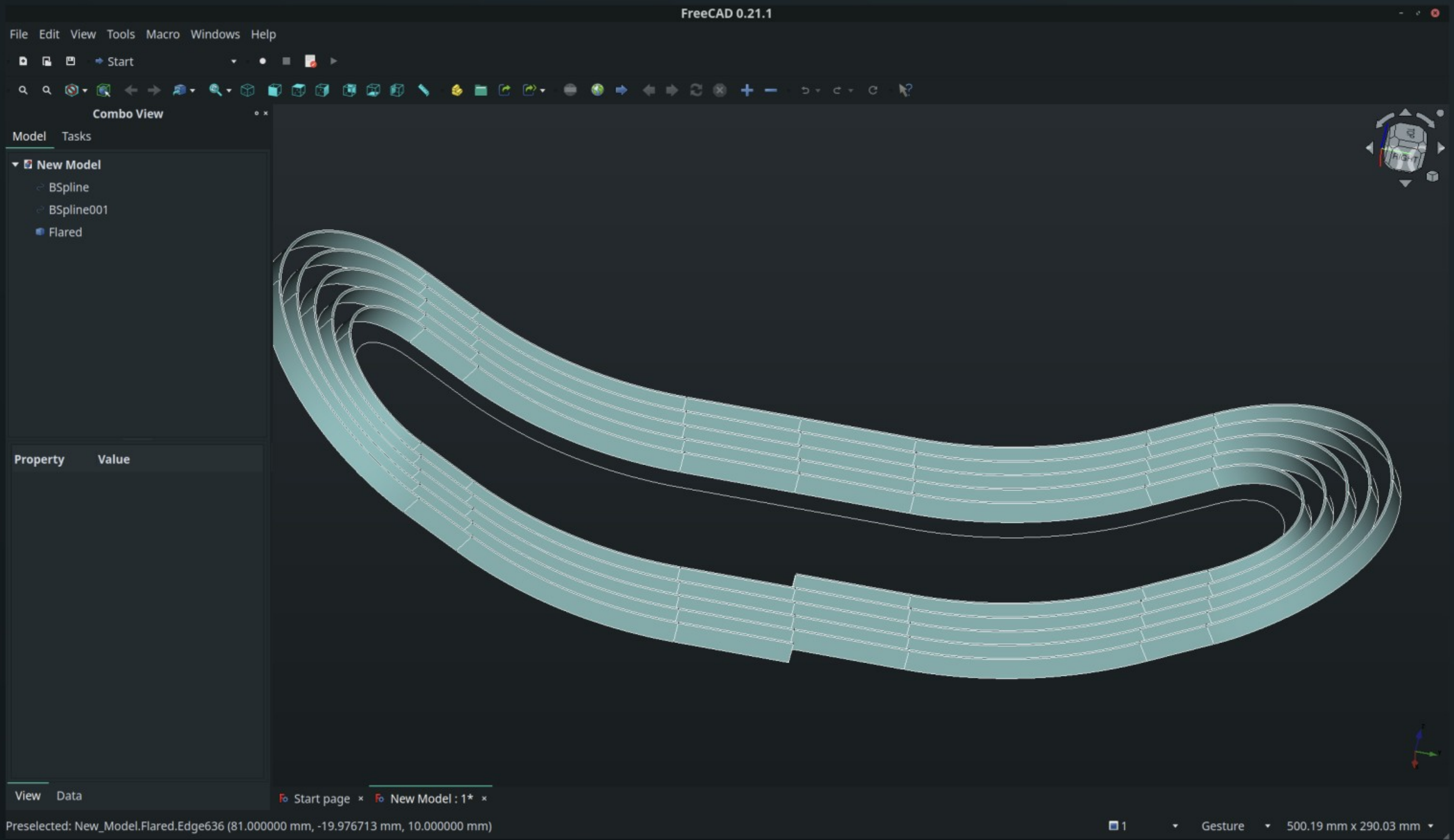
Connectors



- Make hard-way bend free connection between two paths
- Achieved by combining Bezier splines and Frenet-Serret frames
- The control points for the spline are optimized by minimizing the strain energy while constraining
  - Length of the spline
  - Minimum bending radius
  - Edge regression at specified width
- The order of the spline determines the complexity

FreeCAD Macro Export

Most coil geometries can be exported to FreeCAD through the use of a .FCMacro (python script) file that can be automatically generated by RAT.

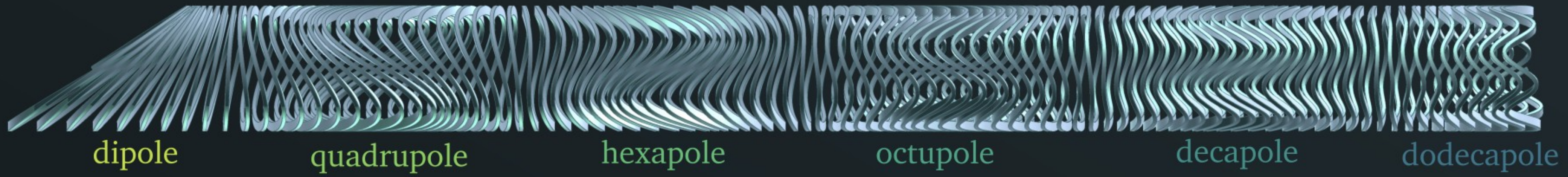


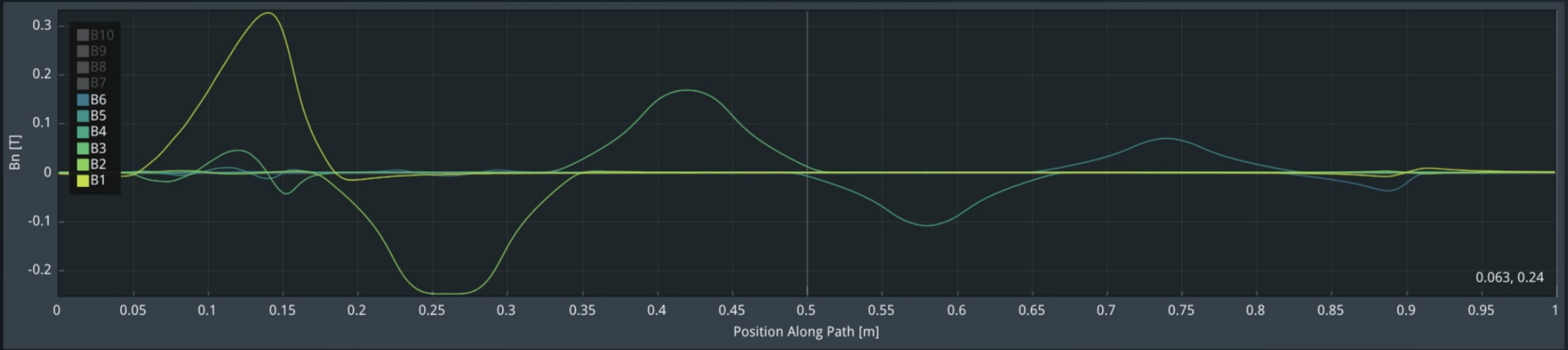
FCMacro File

```python
import FreeCAD,Draft,Part,Import
doc = App.newDocument("New Model")
doc.UndoMode = 0
export=[]
parts=[]
lines=[]
group=[]
# section from ruled surfaces
p0 = FreeCAD.Vector(30.000000000000000,0.000000000000000,0.000000000000000)
p1 = FreeCAD.Vector(30.000000000000000,2.000000000000000,0.000000000000000)
p2 = FreeCAD.Vector(30.000000000000000,4.000000000000000,0.000000000000000)
p3 = FreeCAD.Vector(30.000000000000000,6.000000000000000,0.000000000000000)
p4 = FreeCAD.Vector(30.000000000000000,8.000000000000000,0.000000000000000)
p5 = FreeCAD.Vector(30.000000000000000,10.000000000000000,0.000000000000000)
p6 = FreeCAD.Vector(30.000000000000000,12.000000000000000,0.000000000000000)
p7 = FreeCAD.Vector(30.000000000000000,14.000000000000000,0.000000000000000)
p8 = FreeCAD.Vector(30.000000000000000,16.000000000000000,0.000000000000000)
p9 = FreeCAD.Vector(30.000000000000000,18.000000000000036,0.000000000000000)
p10 = FreeCAD.Vector(30.000000000000000,20.000000000000000,0.000000000000000)
p11 = FreeCAD.Vector(30.000000000000000,22.000000000000000,0.000000000000000)
p12 = FreeCAD.Vector(30.000000000000000,24.000000000000000,0.000000000000000)
p13 = FreeCAD.Vector(30.000000000000000,26.000000000000036,0.000000000000000)
p14 = FreeCAD.Vector(30.000000000000000,28.000000000000000,0.000000000000000)
p15 = FreeCAD.Vector(30.000000000000000,30.000000000000000,0.000000000000000)
p16 = FreeCAD.Vector(30.000000000000000,32.000000000000000,0.000000000000000)
p17 = FreeCAD.Vector(30.000000000000000,34.000000000000000,0.000000000000000)
p18 = FreeCAD.Vector(30.000000000000000,36.000000000000071,0.000000000000000)
p19 = FreeCAD.Vector(30.000000000000000,38.000000000000000,0.000000000000000)
p20 = FreeCAD.Vector(30.000000000000000,40.000000000000000,0.000000000000000)
p21 = FreeCAD.Vector(30.000000000000000,42.000000000000000,0.000000000000000)
p22 = FreeCAD.Vector(30.000000000000000,44.000000000000000,0.000000000000000)
p23 = FreeCAD.Vector(30.000000000000000,46.000000000000000,0.000000000000000)
p24 = FreeCAD.Vector(30.000000000000000,48.000000000000000,0.000000000000000)
p25 = FreeCAD.Vector(30.000000000000000,50.000000000000000,0.000000000000000)
spline0 = Draft.make_bspline([p0,p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p1
spline0.ViewObject.LineColor = (1.0,1.0,1.0,0.0)
spline0.ViewObject.LineWidth = 0.5
lines.append(spline0)
# section from ruled surfaces
p0 = FreeCAD.Vector(30.000000000000000,50.000000000000000,0.000000000000000)
p1 = FreeCAD.Vector(30.000000000000000,51.975812296293142,0.009759823710853)
p2 = FreeCAD.Vector(30.000000000000000,53.951431756789304,0.039038342301833)
p3 = FreeCAD.Vector(30.000000000000000,55.926665564511949,0.087832698241130)
p4 = FreeCAD.Vector(30.000000000000000,57.901320940123568,0.156138129285632)
p5 = FreeCAD.Vector(30.000000000000000,59.875205160740598,0.243947968945668)
p6 = FreeCAD.Vector(30.000000000000000,61.848125578742866,0.351253647135707)
p7 = FreeCAD.Vector(30.000000000000000,63.819889640575596,0.478044691010737)
p8 = FreeCAD.Vector(30.000000000000000,65.790304905542370,0.624308725988354)
p9 = FreeCAD.Vector(30.000000000000000,67.759179064586959,0.790031476956620)
p10 = FreeCAD.Vector(30.000000000000000,69.726319959062351,0.975196769667174)
p11 = FreeCAD.Vector(30.000000000000000,71.691535599485106,1.179786532313898)
p12 = FreeCAD.Vector(30.000000000000000,73.654634184273163,1.403780797296616)
p13 = FreeCAD.Vector(30.000000000000000,75.615424118465384,1.647157703169921)
p14 = FreeCAD.Vector(30.000000000000000,77.573714032420866,1.909893496776882)
p15 = FreeCAD.Vector(30.000000000000000,79.529312800496242,2.191962535567193)
p16 = FreeCAD.Vector(30.000000000000000,81.482029559699270,2.493337290099951)
p17 = FreeCAD.Vector(30.000000000000000,83.431673728316667,2.813988346730389)
p18 = FreeCAD.Vector(30.000000000000000,85.378055024514594,3.153884410480634)
p19 = FreeCAD.Vector(30.000000000000000,87.320983484909803,3.512992308094098)
p20 = FreeCAD.Vector(30.000000000000000,89.260269483109709,3.891276991273007)
p21 = FreeCAD.Vector(30.000000000000000,91.195723748219677,4.288701540099193)
p22 = FreeCAD.Vector(30.000000000000000,93.127157383315392,4.705227166637283)
p23 = FreeCAD.Vector(30.000000000000000,95.054381883879003,5.140813218720442)
p24 = FreeCAD.Vector(30.000000000000000,96.977209156196735,5.595417183917851)
p25 = FreeCAD.Vector(30.000000000000000,98.895451535716532,6.068994693683916)
p26 = FreeCAD.Vector(30.000000000000000,100.808921805363823,6.561499527688585)
p27 = FreeCAD.Vector(30.000000000000000,102.717433213813521,7.072883618328251)
p28 = FreeCAD.Vector(30.000000000000000,104.620799493716660,7.603097055417240)
p29 = FreeCAD.Vector(30.000000000000000,106.518834879879776,8.152088091058740)
p30 = FreeCAD.Vector(30.000000000000000,108.411354127395171,8.719803144695516)
p31 = FreeCAD.Vector(30.000000000000000,110.298172529720588,9.306186808339145)
p32 = FreeCAD.Vector(30.000000000000000,112.179105936706264,9.911181851977792)
```

Custom CCT

In a Custom CCT Path the harmonic strength An and Bn can be set along the length of the coil.
A superposition of multiple harmonics at the same axial position is possible as well.





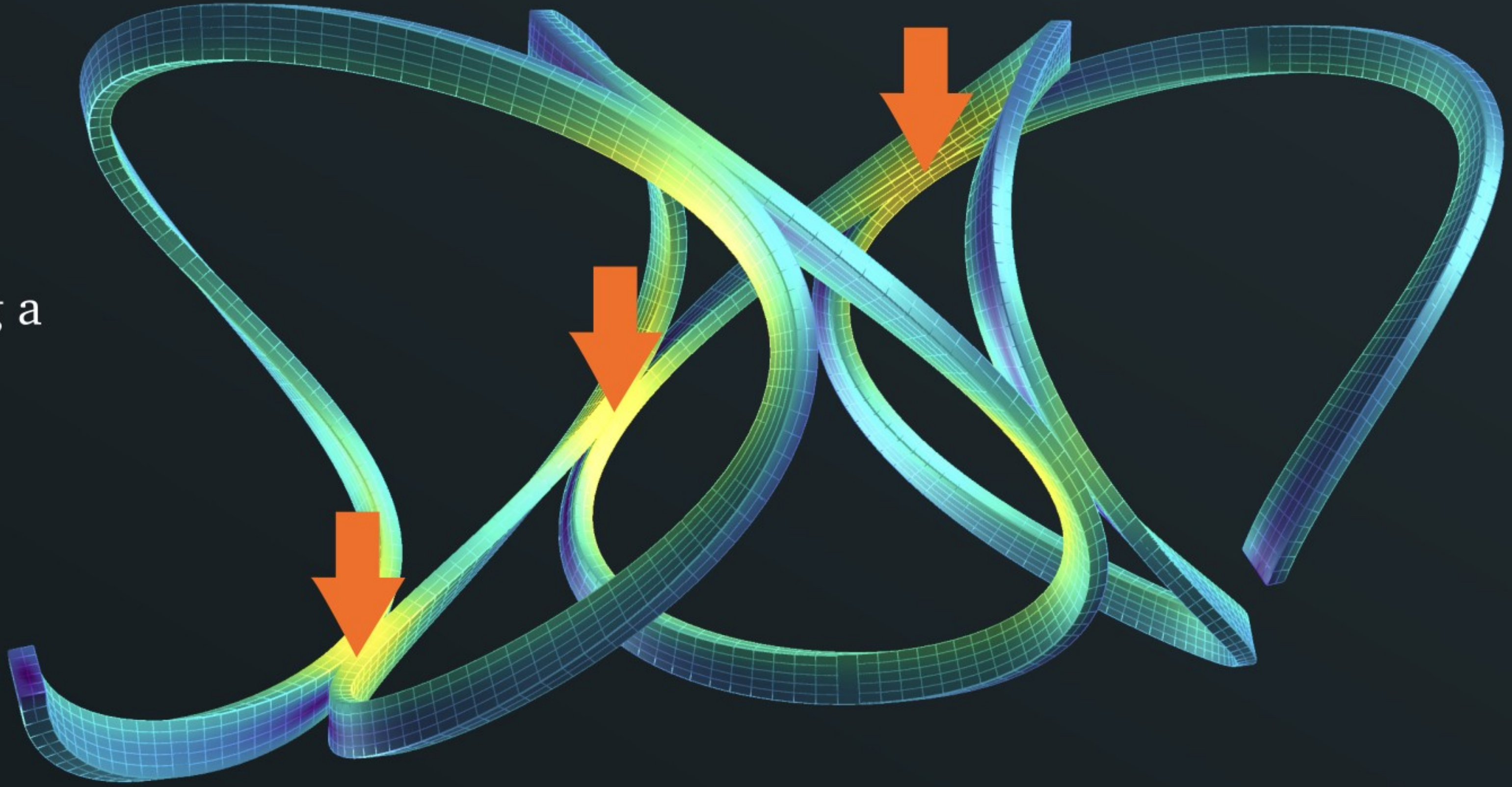
Controlling CCT Winding Pitch

When using variable harmonics along the length of the magnet we would like to maintain positive spacing between turns.

First the path is generated with zero pitch causing all turns to coincide

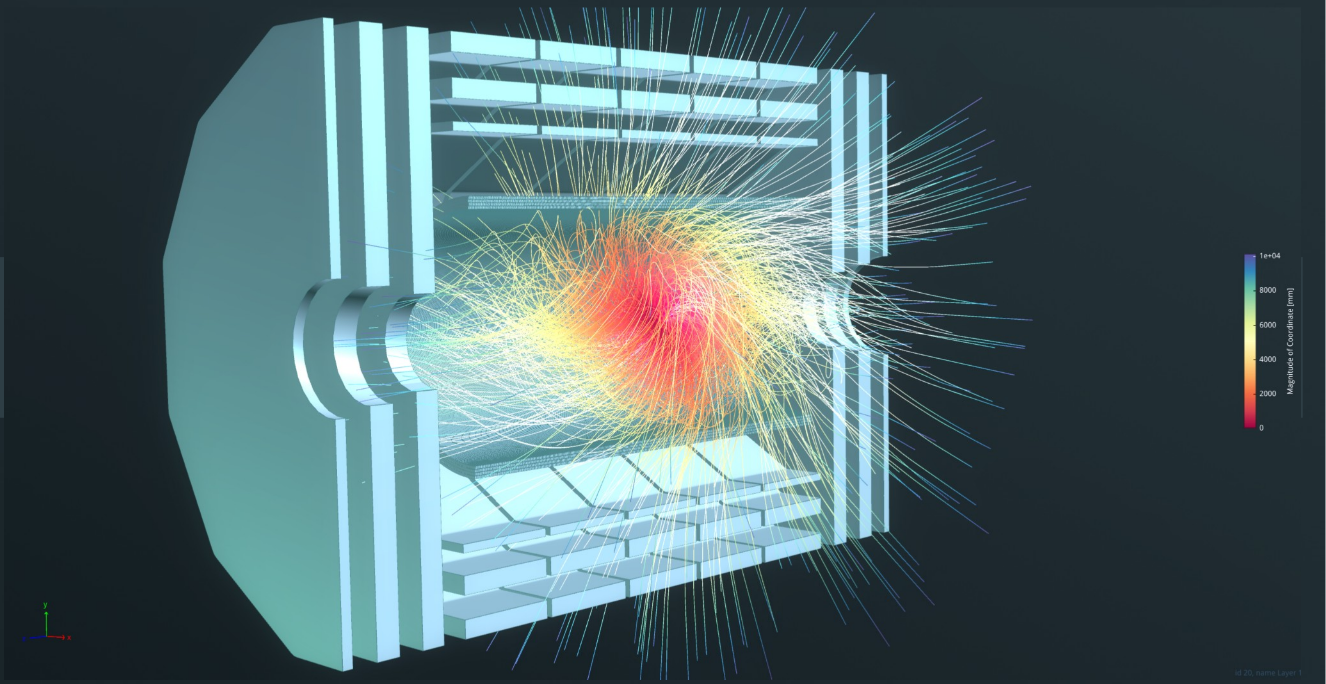
Then the local pitch is determined by using a window, scanning one turn ahead and one turn behind to find minimum spacing between turns

The pitch then must be numerically integrated to get the z-offset



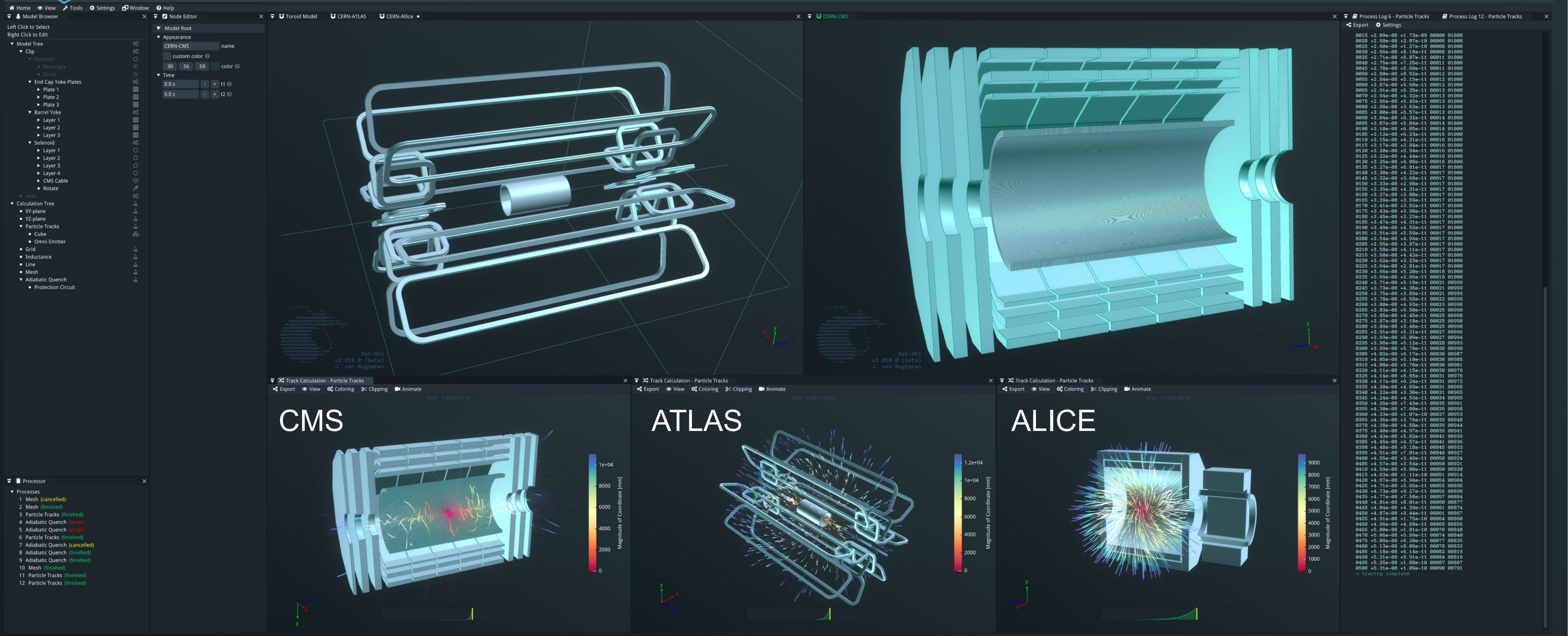
CMS Example

Modeling CMS Magnet System in RAT



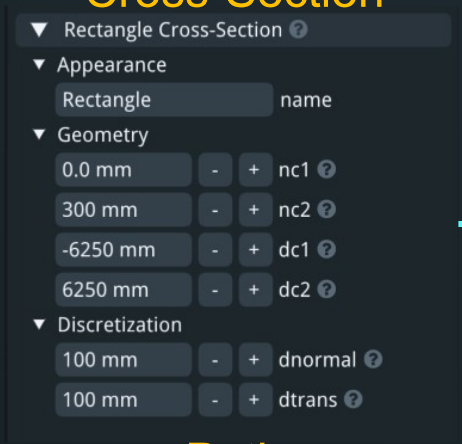
Note: CMS coils modeled here are a loose representation based on info found on internet ...

Other CERN Detector Magnets

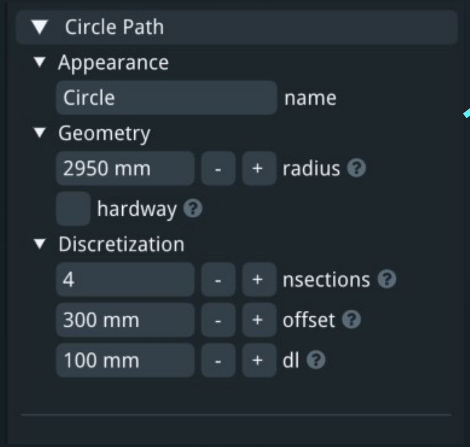


Step 1: Modeling the CMS solenoid

Cross-Section

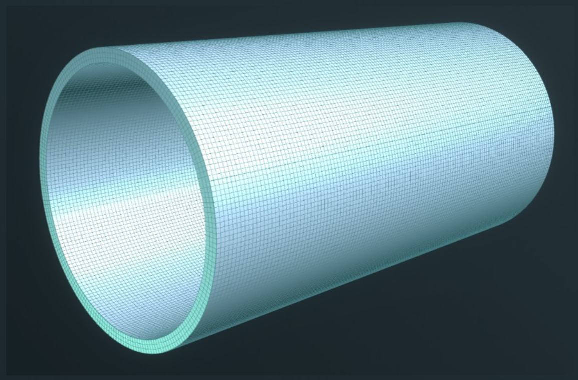


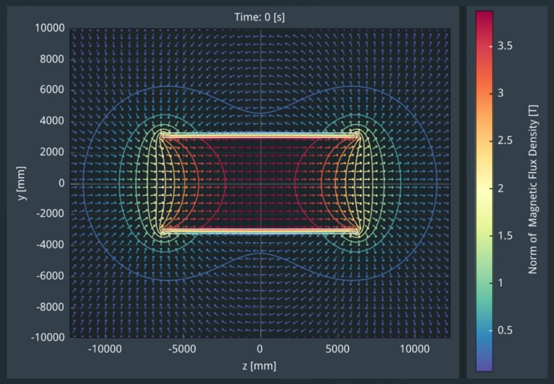
Path



Custom Coil

=



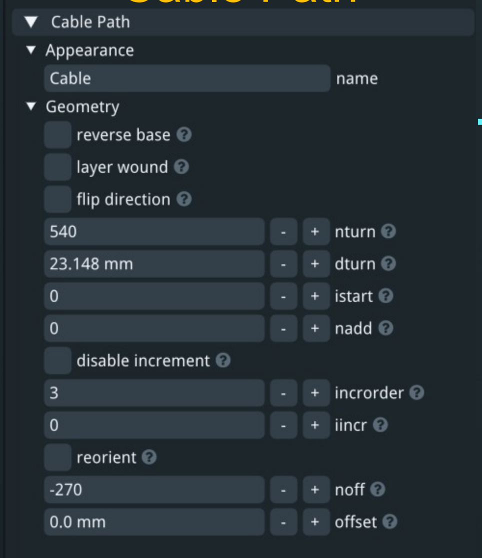


- Use a Circle Path
  - Radius 2950 mm
- Use a Rectangle Cross Section
  - From 0 to 300 mm in the normal direction
  - From -6250 to +6250 in the transverse direction
- Element sizes set to 100 mm

Step 2: Adding Turn Detail

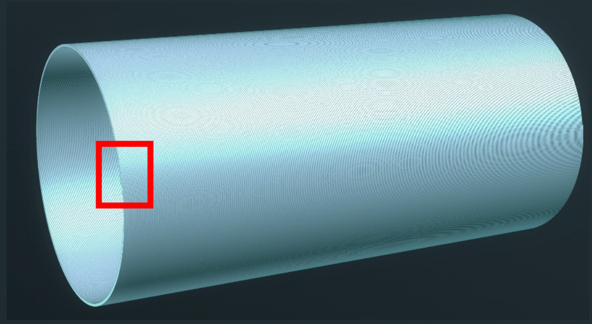
Circle Path

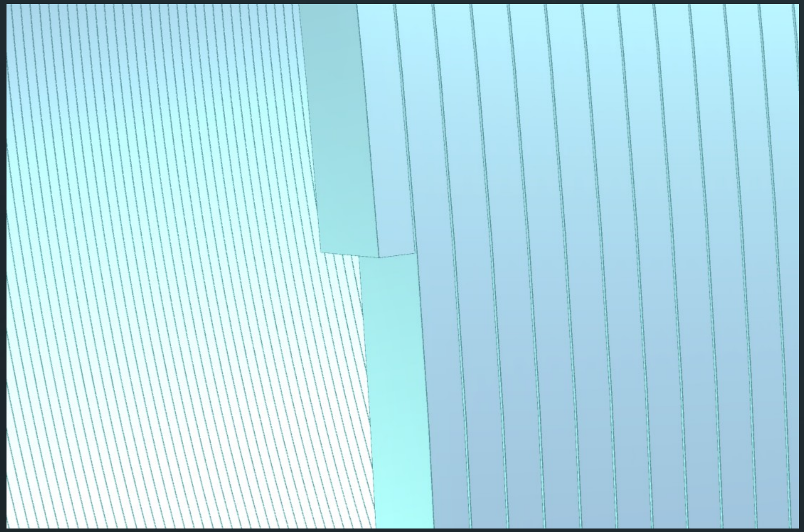
Cable Path



Custom Coil

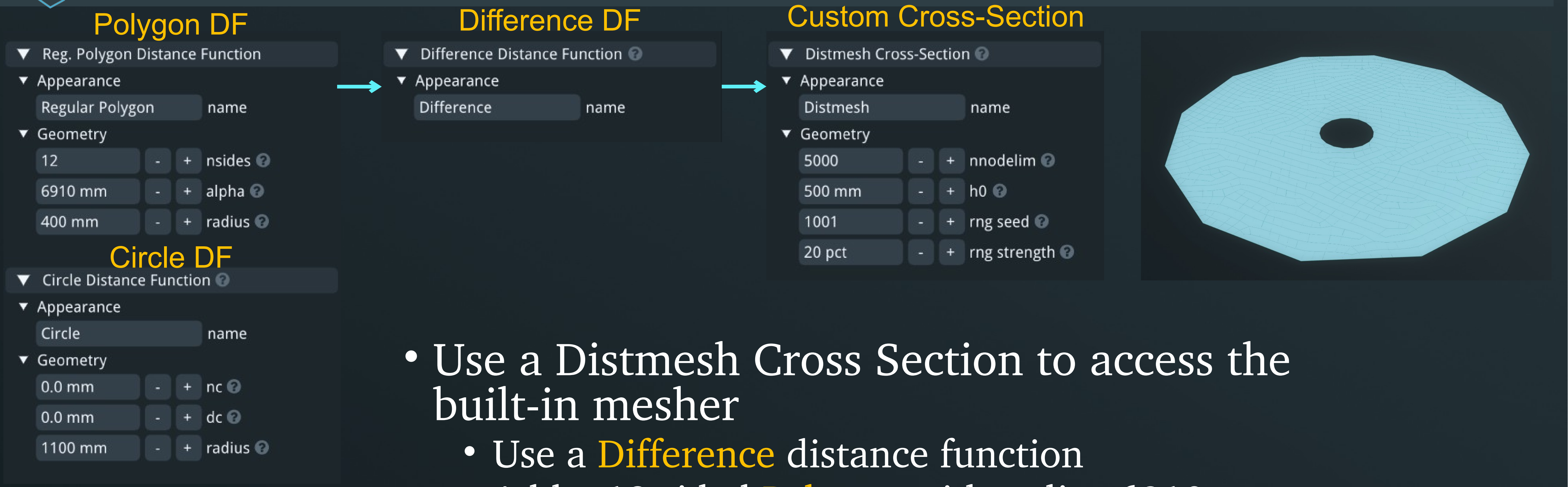
=





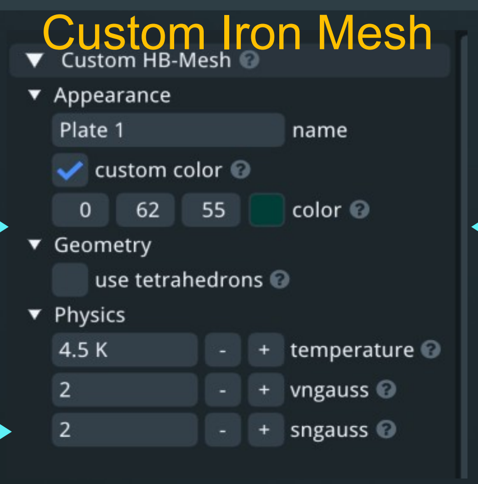
- Add the Circle Path to a Cable Path to make individual coil windings
- The cross section is now set to the cross section of a cable
- The number of turns is now set to 1
- Other layers are made using increasing circle radii and alternating reverse base

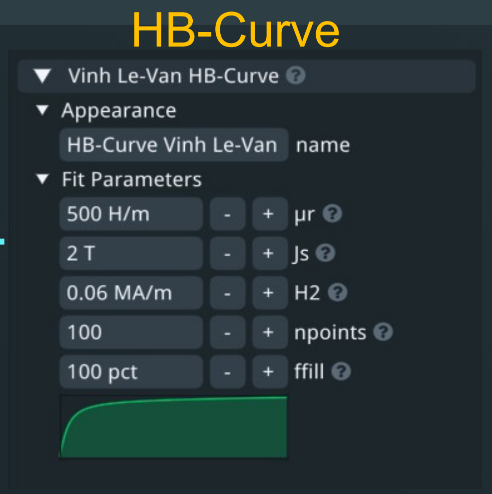
Step 3: End Cap Yoke Plate Cross Section

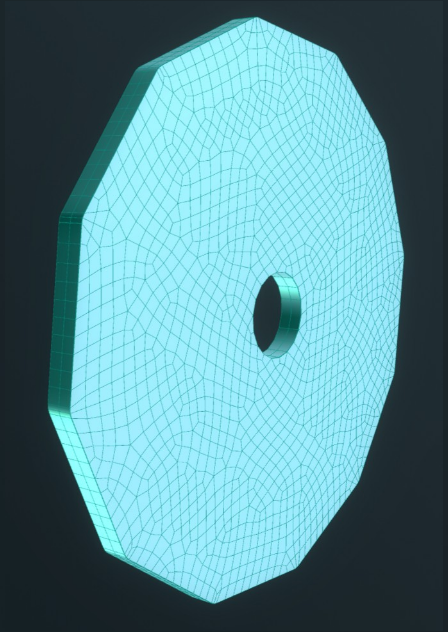


- Use a Distmesh Cross Section to access the built-in mesher
  - Use a Difference distance function
  - Add a 12 sided Polygon with radius 6910 mm
  - Subtract a Circle with radius 1100, 1300 and 1500 mm for the plates respectively
  - Element size 500 mm

Step 4: Extrude the Mesh along Axis





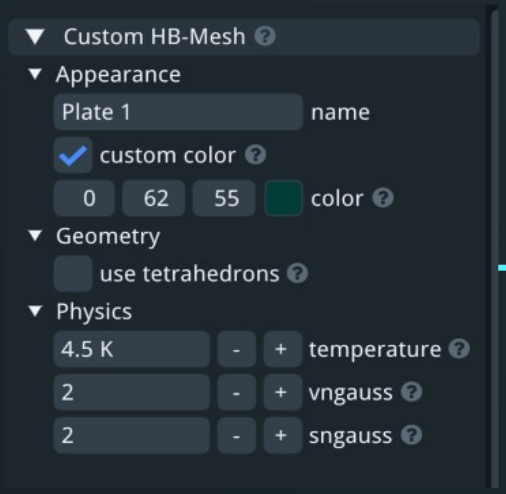


- Use a Custom HB Mesh
  - Add the Distmesh Cross Section that we created in the previous step
  - Add and an axial path along the z-axis of length 600 mm
  - Add an HB curve fit with saturation of 2 T
- Other end-plates are extruded in a similar way

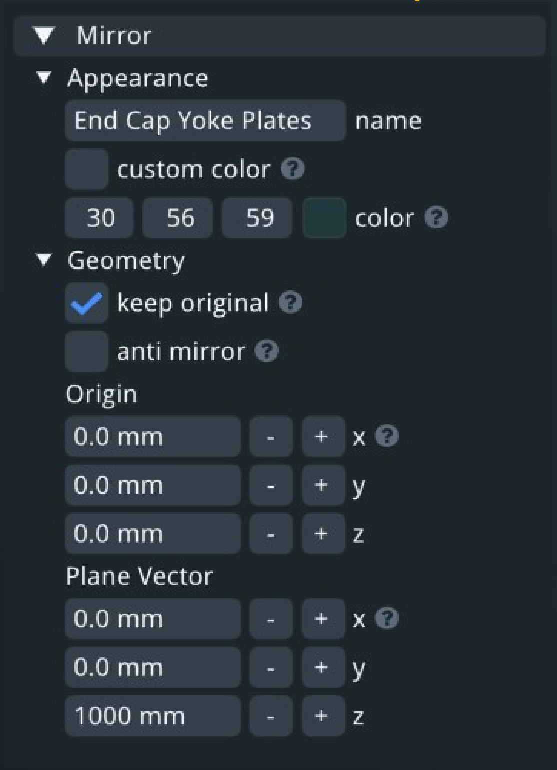
Step 5: Mirror the Yoke's End Cap Plate

Custom Iron Mesh

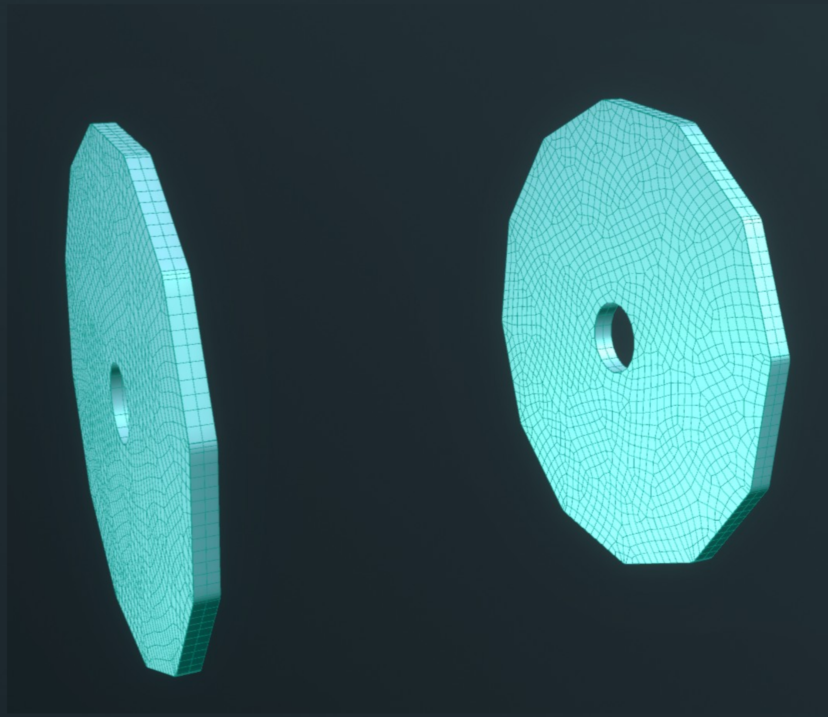
…



Mirror Group



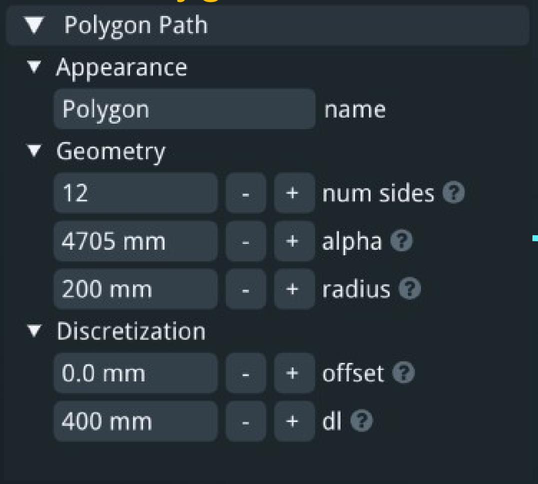
=



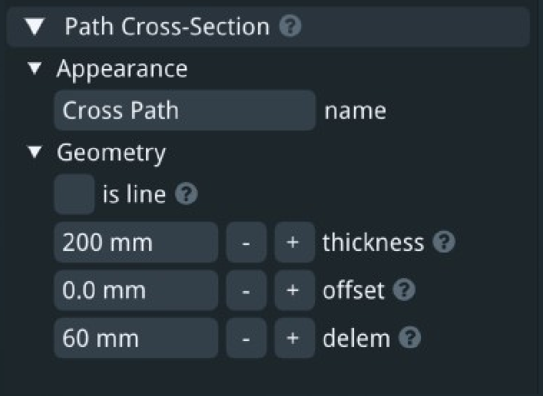
- Using a Mirror along the z-axis the end-plate is duplicated to the other side
- All end-plates can be added to the same mirror object

Step 6: Section of barrel yoke

Polygon Path



Path Cross-Section



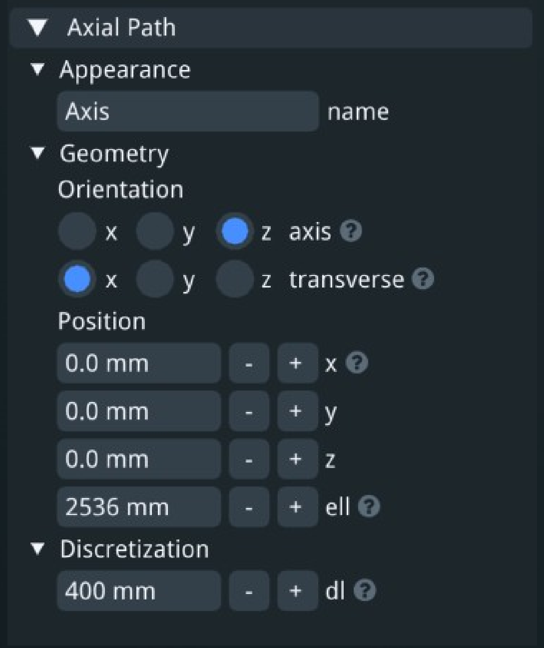
=

- We could subtract two 12 sided polygons and use the distmesher again
- More regular mesh can be achieved with Path Cross Section, which creates a cross section from a path by thickening it
- Other layers made similarly

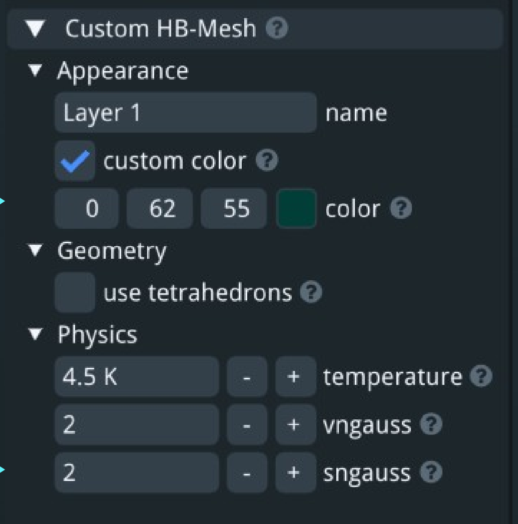
Step 7: Extrude the mesh

Path Cross-Section

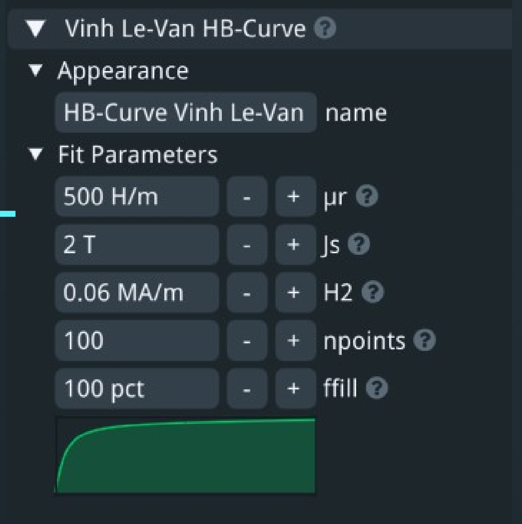
Axial Path

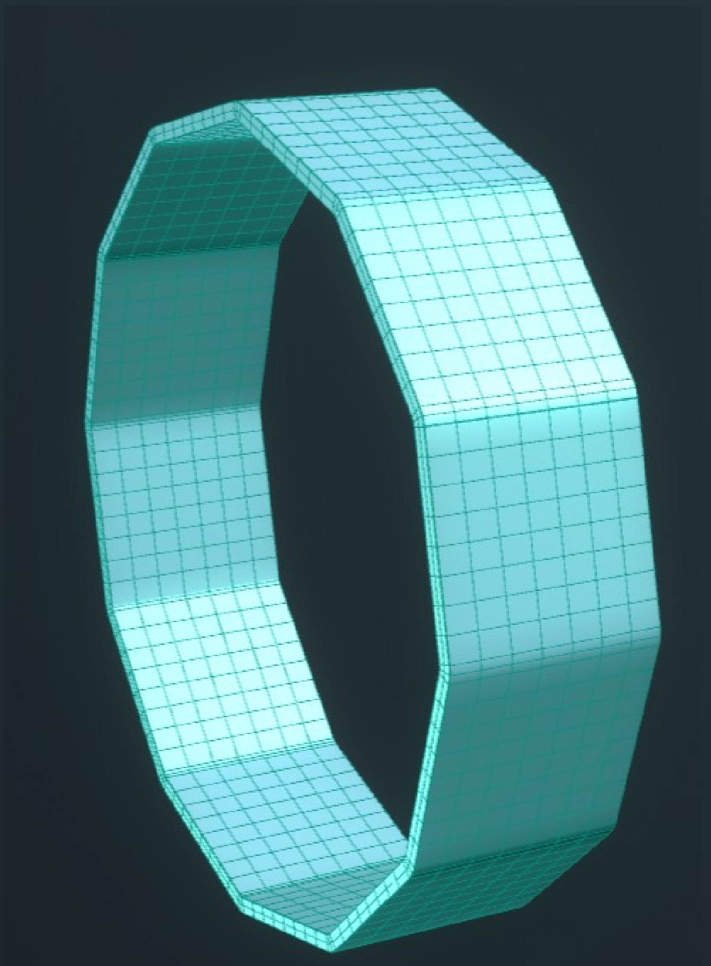


Custom Iron Mesh



HB-Curve

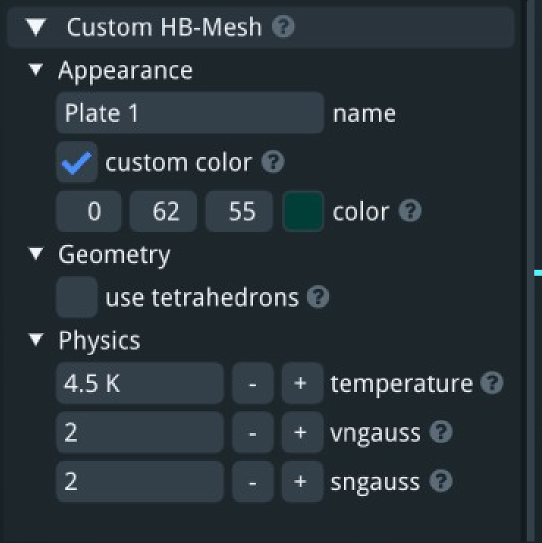




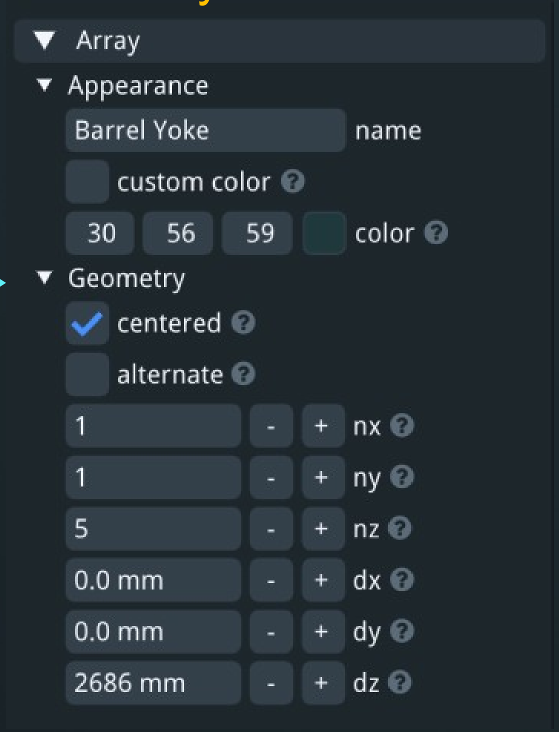
- Use a Custom HB Mesh
  - Add the Path based Cross Section that we created in the previous step
  - Add and an axial path along the z-axis of length 2536 mm
  - Add an HB curve fit with saturation of 2 T
- Other layers are extruded in a similar way

Step 8: array of Barrel yoke section

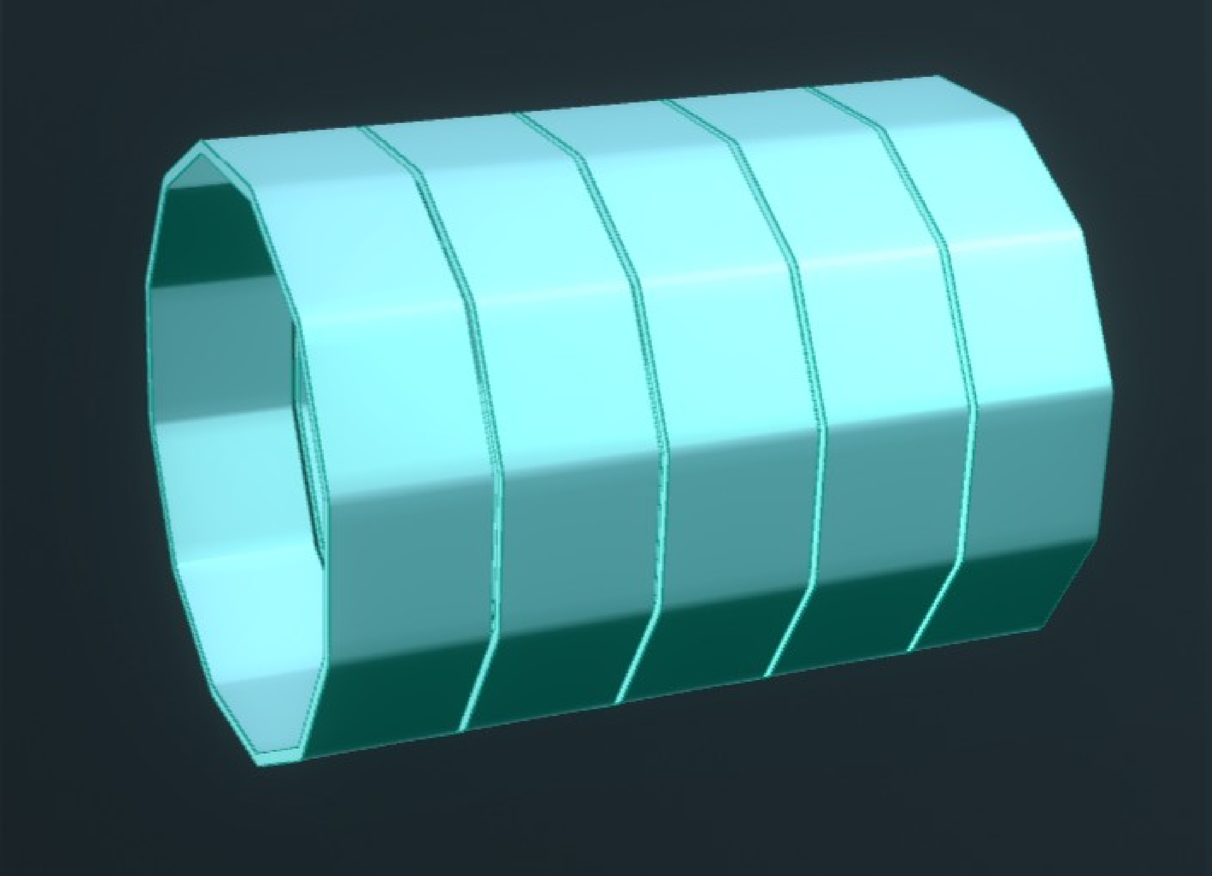
Custom Iron Mesh



Array of Meshes

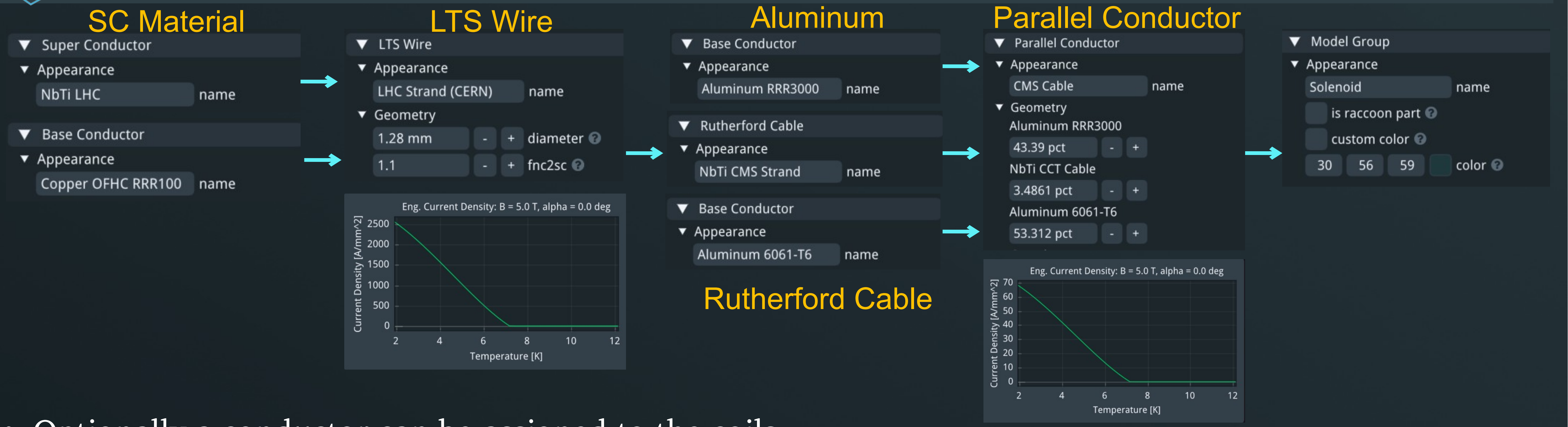


=



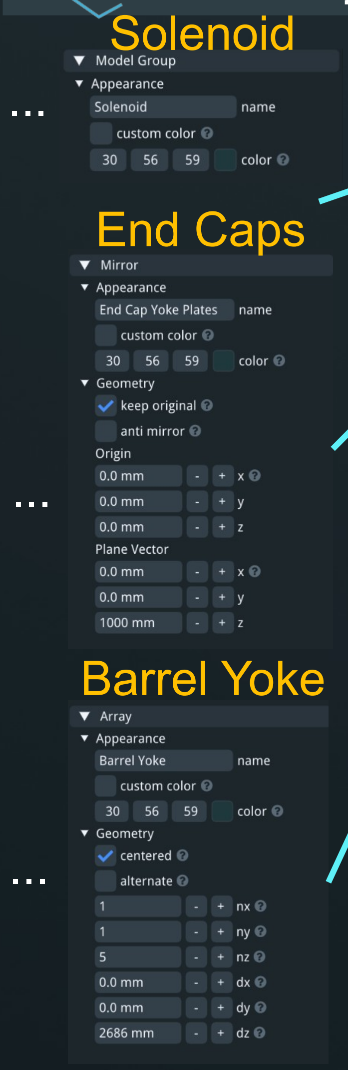
- Using the Array along the z-axis the ring is duplicated 5 times
- All layers are included in the same array

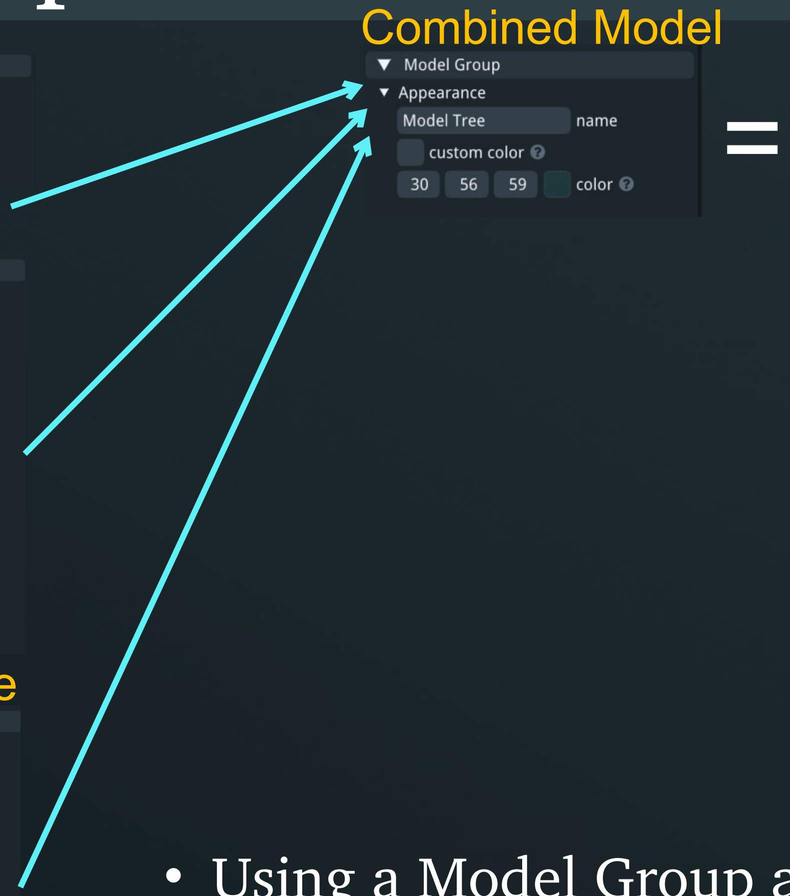
Step 9: Add the CMS Conductor

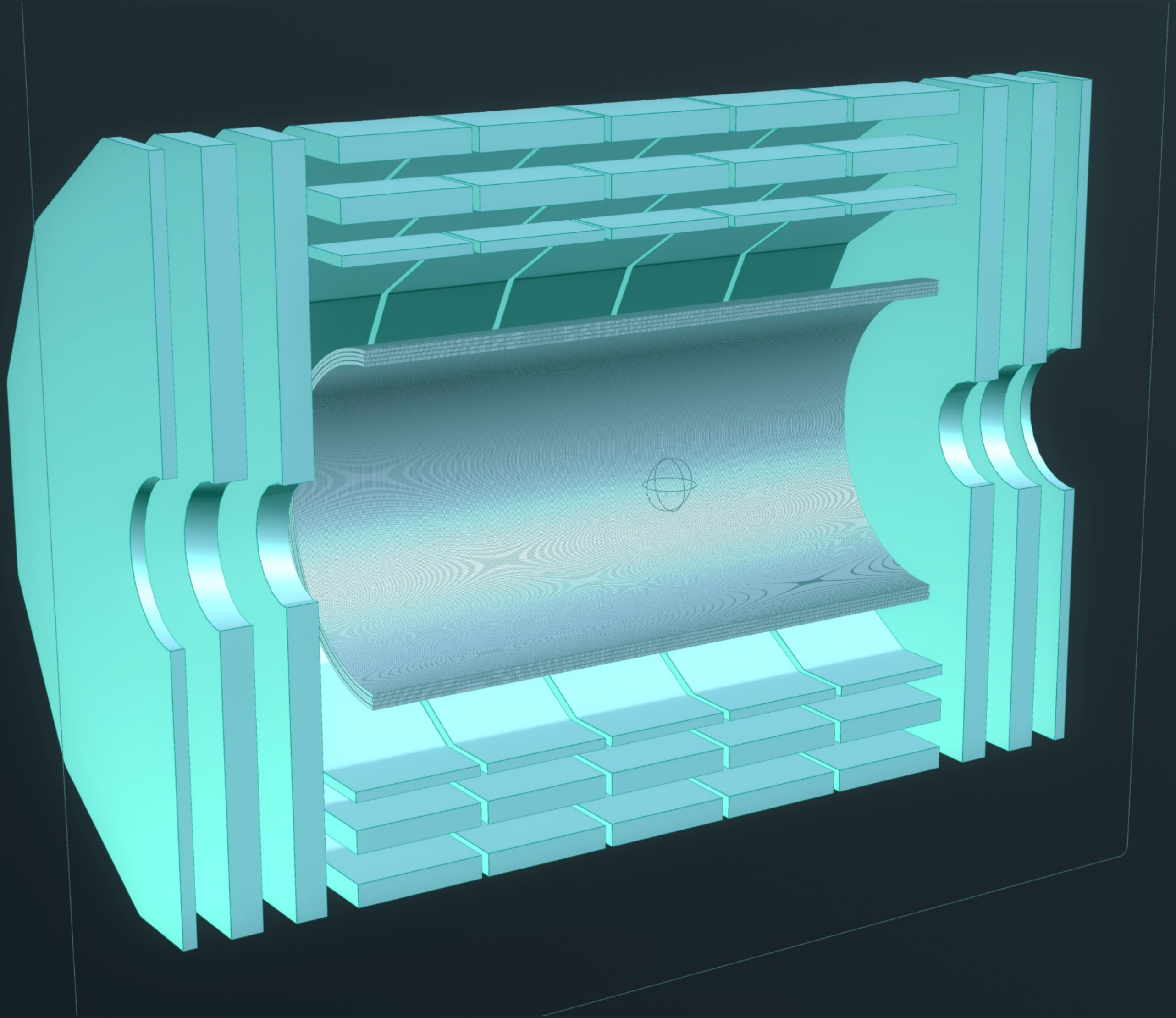


- Optionally a conductor can be assigned to the coils
  - Base conductor has k, rho, Cp and d
  - Superconductor also has Jc and N
- All conductors can be combined using
  - Parallel or Series conductors
  - LTS wire and HTS tape are just wrappers around parallel conductor
- This allows for calculation of margins/critical currnet on coil etc.

Step 10: Combine the Models





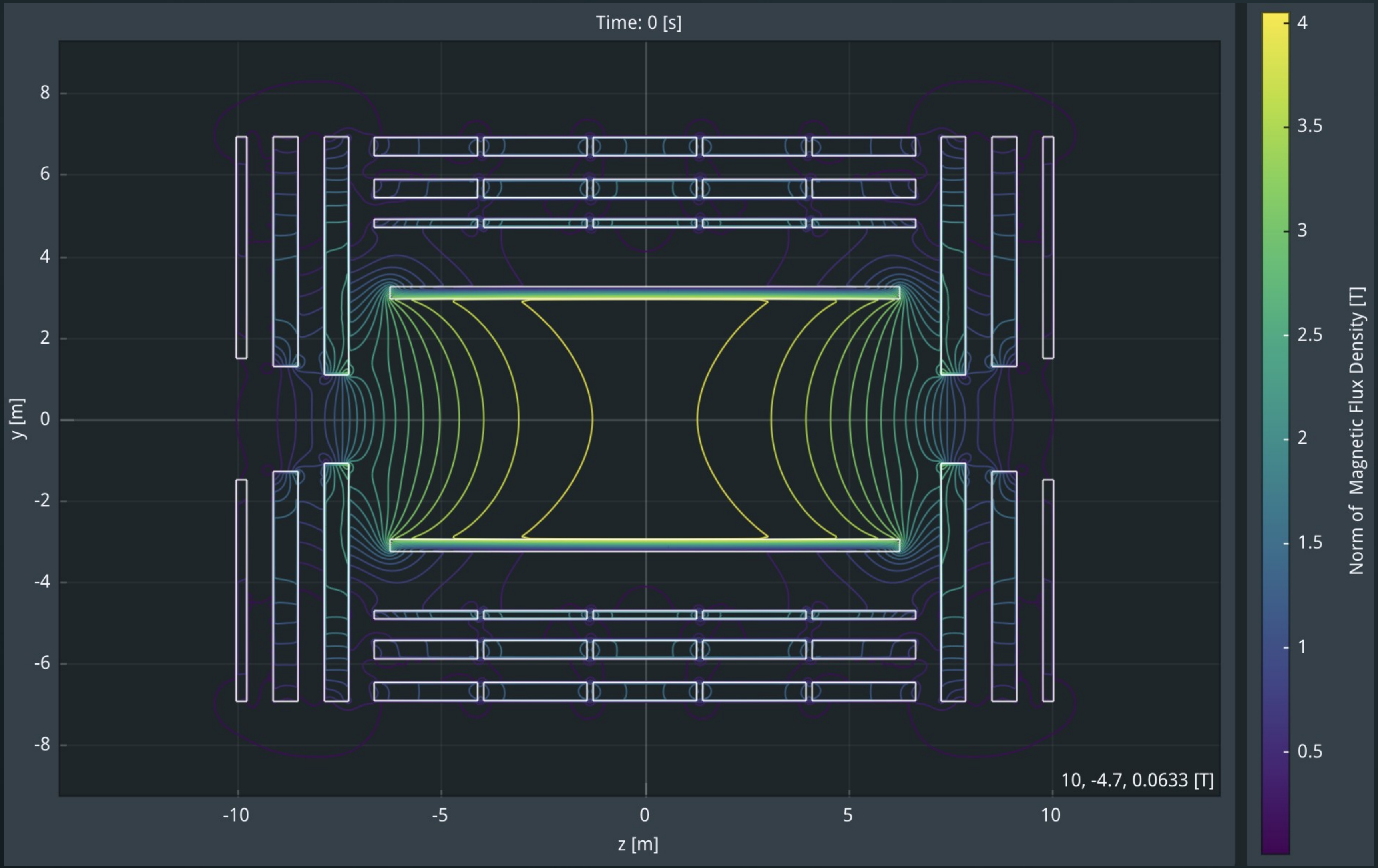


- Using a Model Group all objects can be combined into a single model, ready for calculations

CMS Calculations

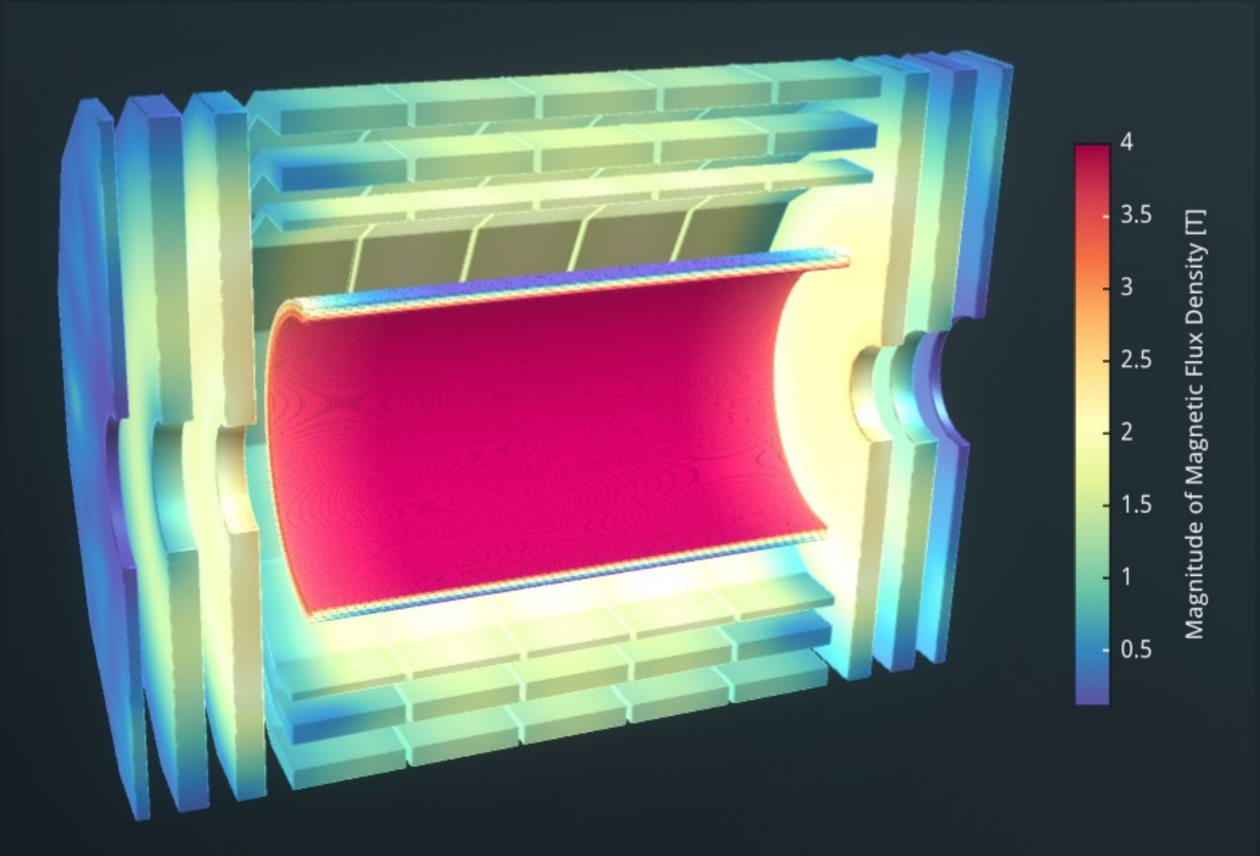
Example calculations on the CMS model

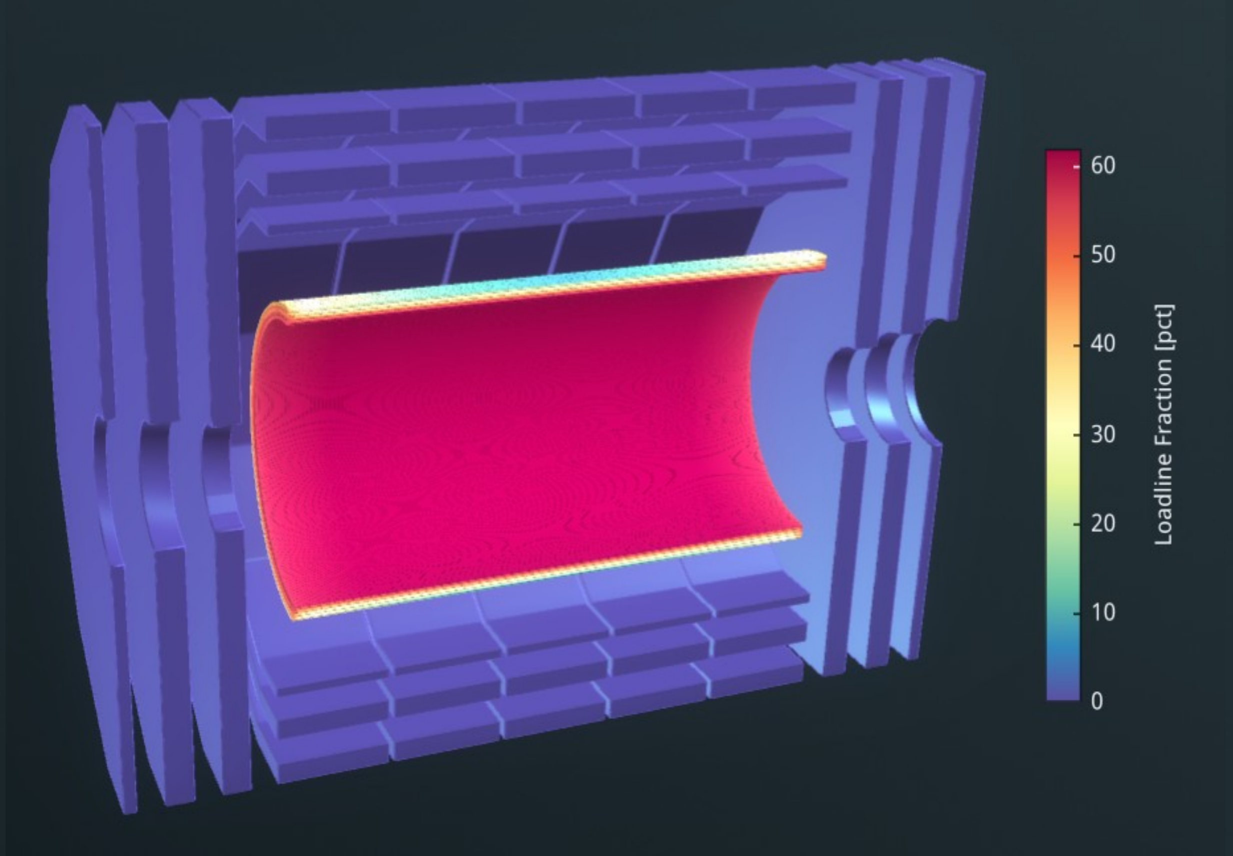
Plane Calculation



- Connect the model to Plane Calculation
- Plane is defined by
  - orientation YZ
  - Size 24 by 18 m
  - Number of pixels 800 by 600
- Field at center 4.0 T

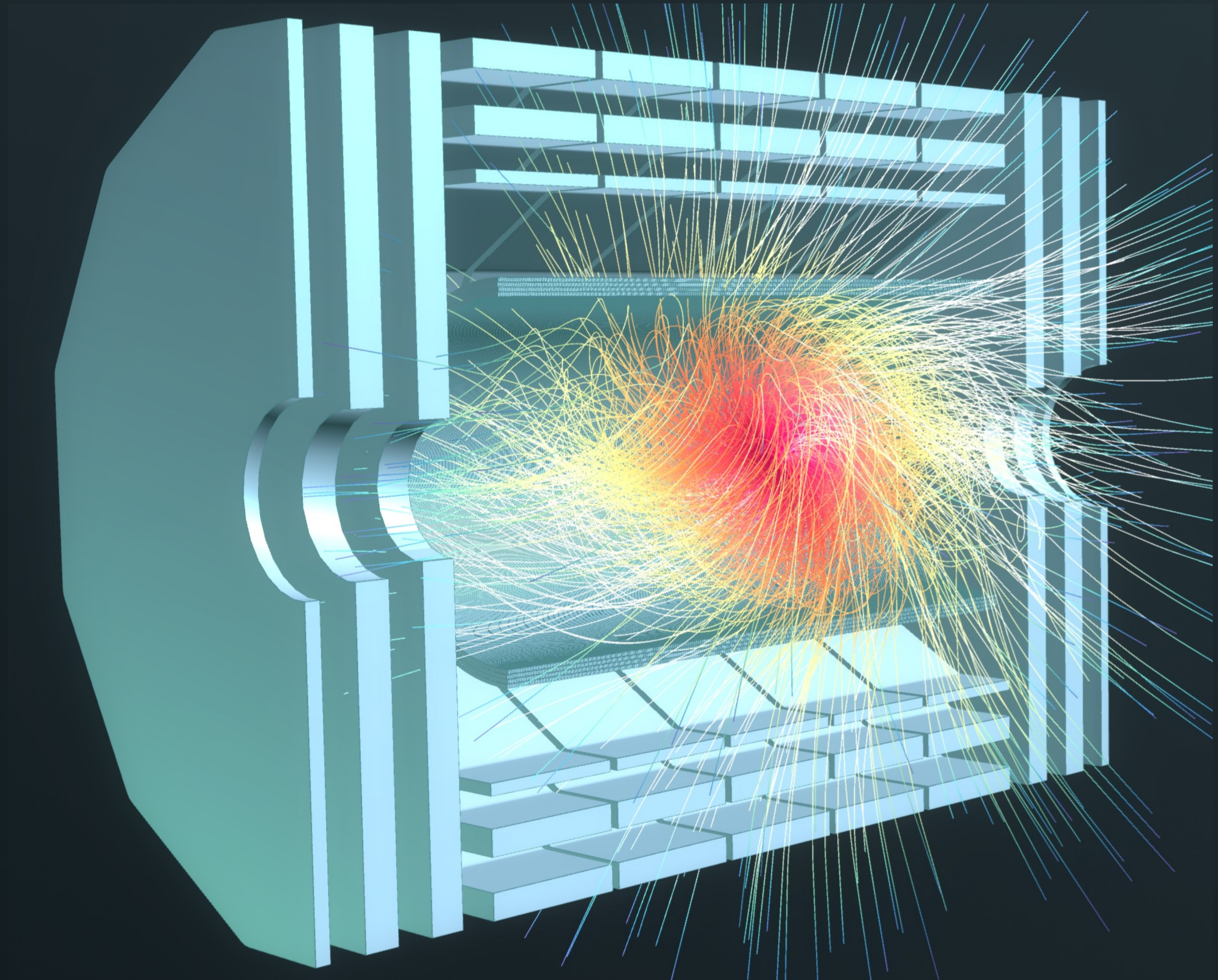
Mesh Calculation





- Mesh calculation allows calculating field on and in conductor
- Because we included the material properties we also get loadline fraction and more

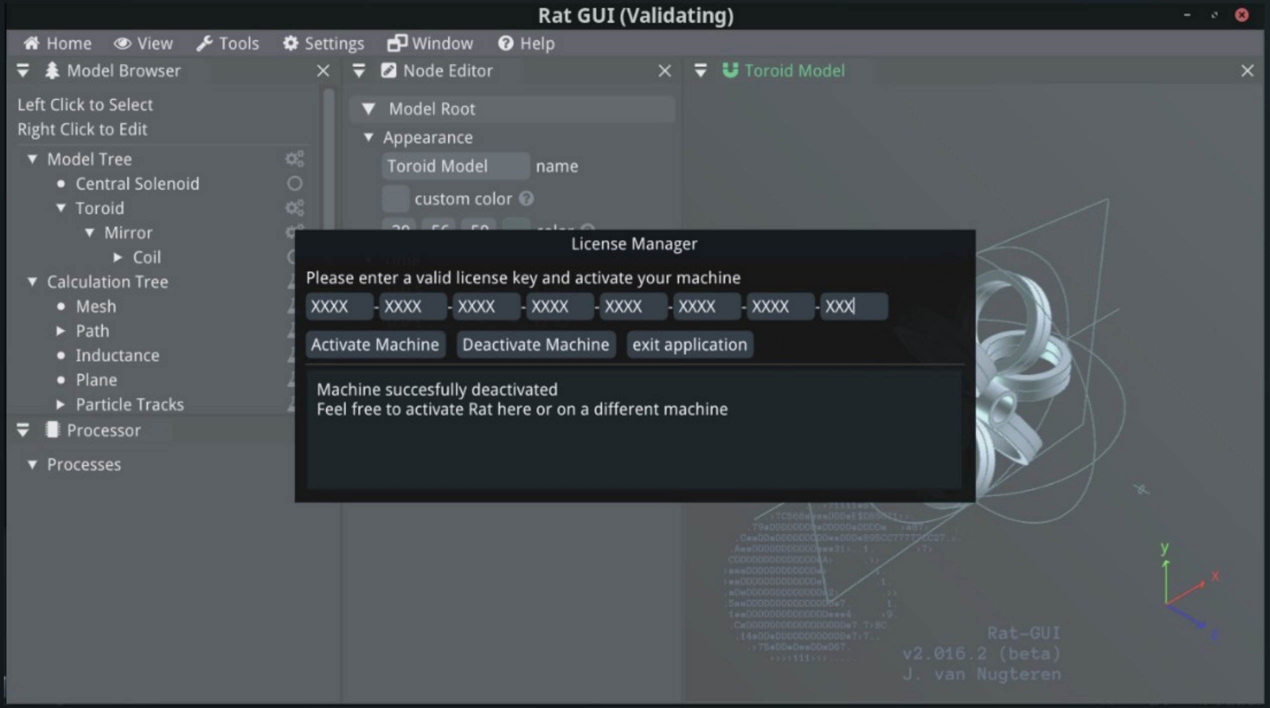
Particle Tracking

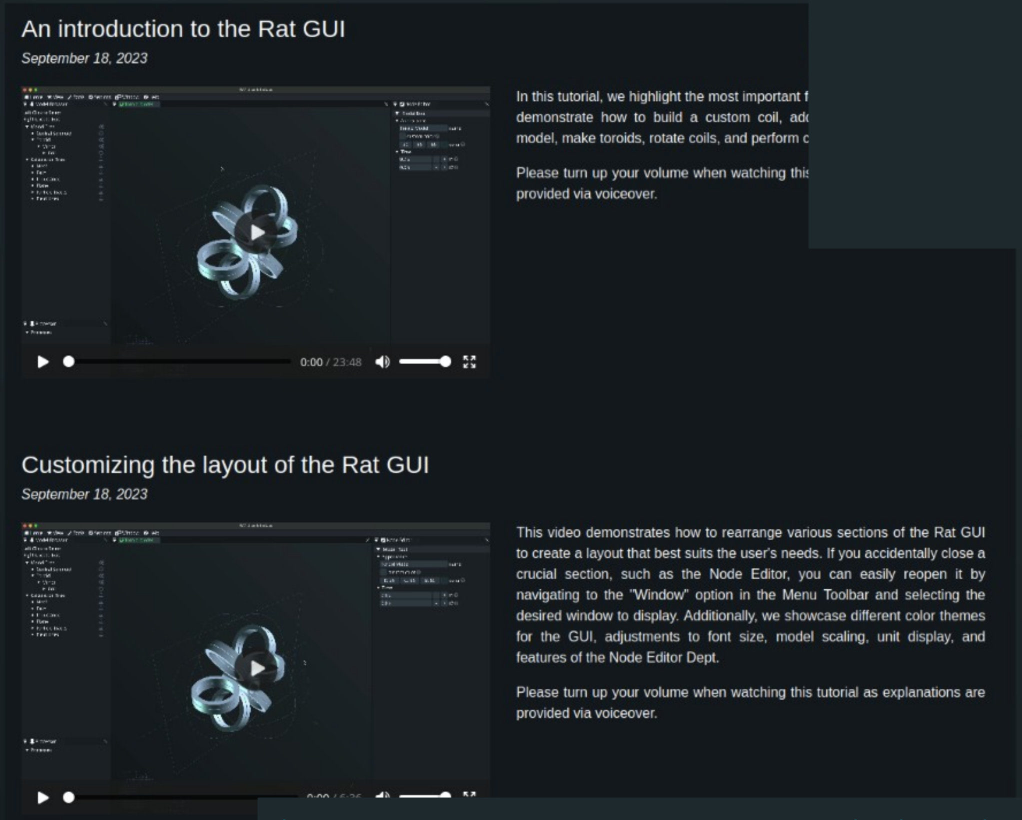


- Uses pre-calculated FMM to calculate the field at the target positions
- Uses B-field to calculate particle trajectory based on
    - rest mass
    - electric charge
    - particle energy / momentum
- Changing B-field causes E-field, this is included in calculations
- Particle interactions are currently not taken into account

Try RAT GUI Today

1. After the presentation ask Nikkie for a trial key or send an e-mail to: info@rat-gui.com
2. Download RAT from: https://rat-gui.com/download.html
3. Install latest RAT on your computer (circumvent microsoft unidentified developer nonsense).
4. Type the key in the XXXX boxes indicated in the screenshot and hit activate.
5. You can use RAT for free for 30 days.
6. The key can be active on one PC at a time.



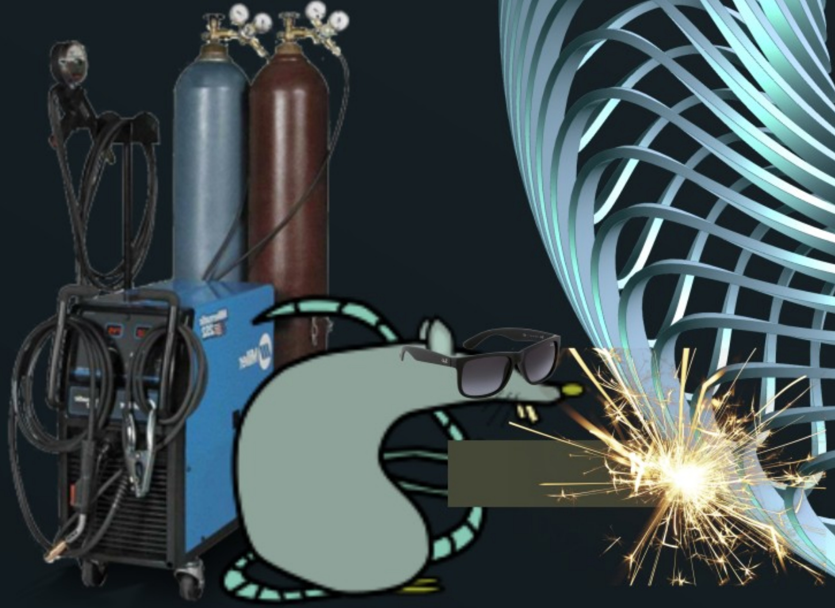


https://rat-gui.com/tutorials.html

For more information and source code:
https://gitlab.com/Project-Rat

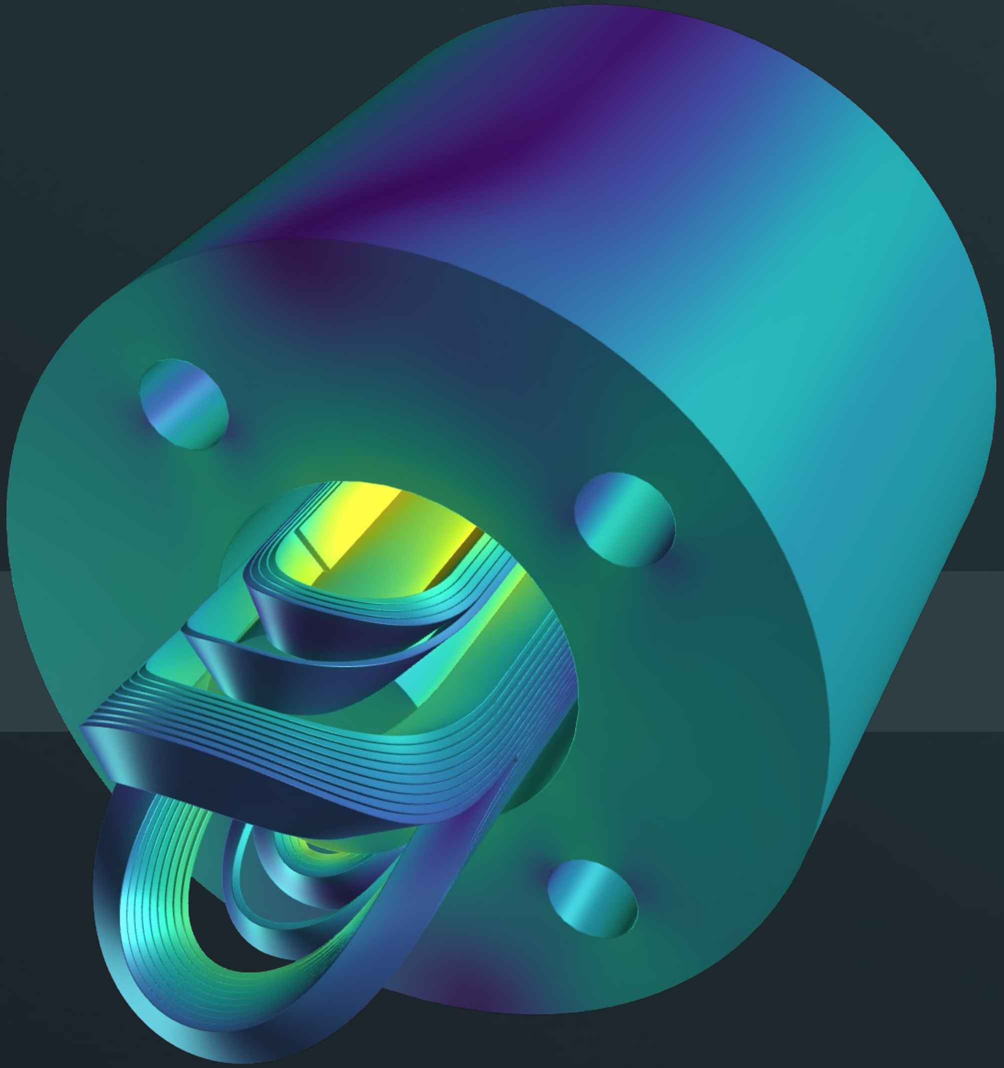
https://rat-gui.com/

Thanks



Little Beast
engineering

Back-up slides



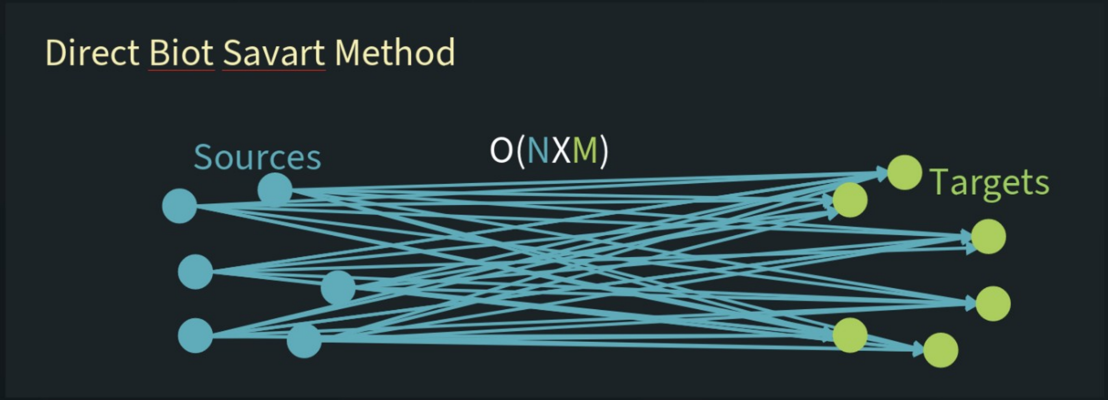
How does ЯΔ⊣ Work?

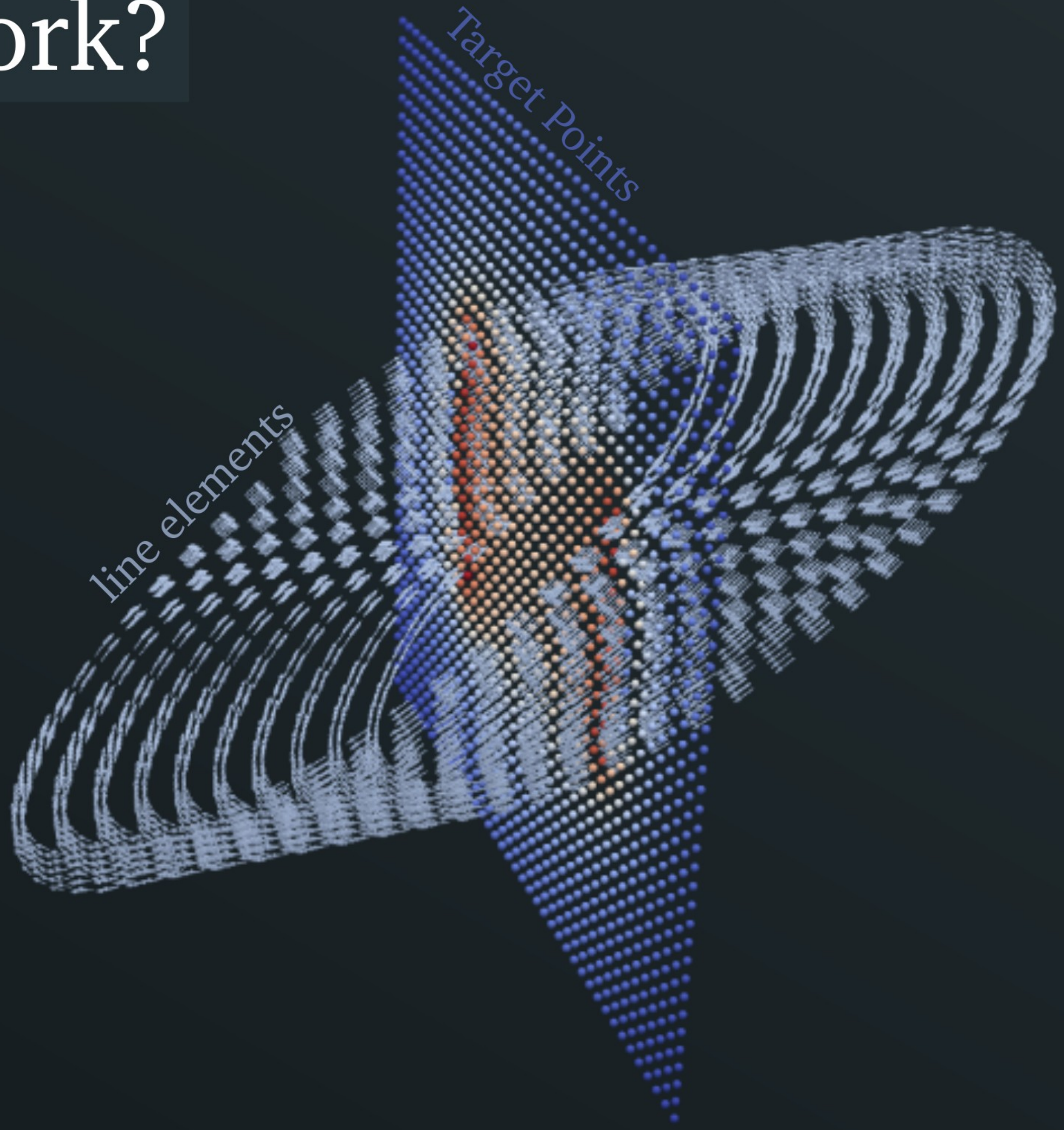
For a magnetic field calculation the coil is split into N line elements

The M target points are defined by the type of calculation, for instance a plane

Integrating directly from the Biot-Savart equation results in O(NXM) complexity

This will take a long time when N and M are large





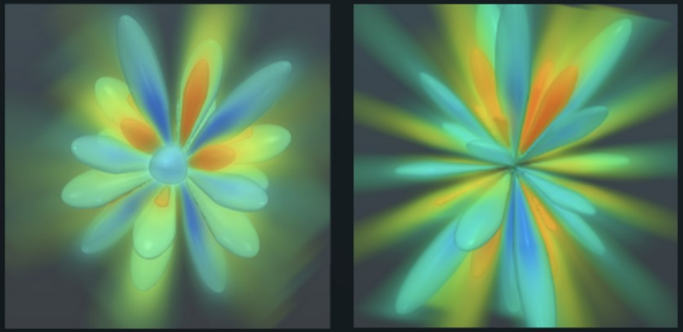
Fast Multipole Method (I)

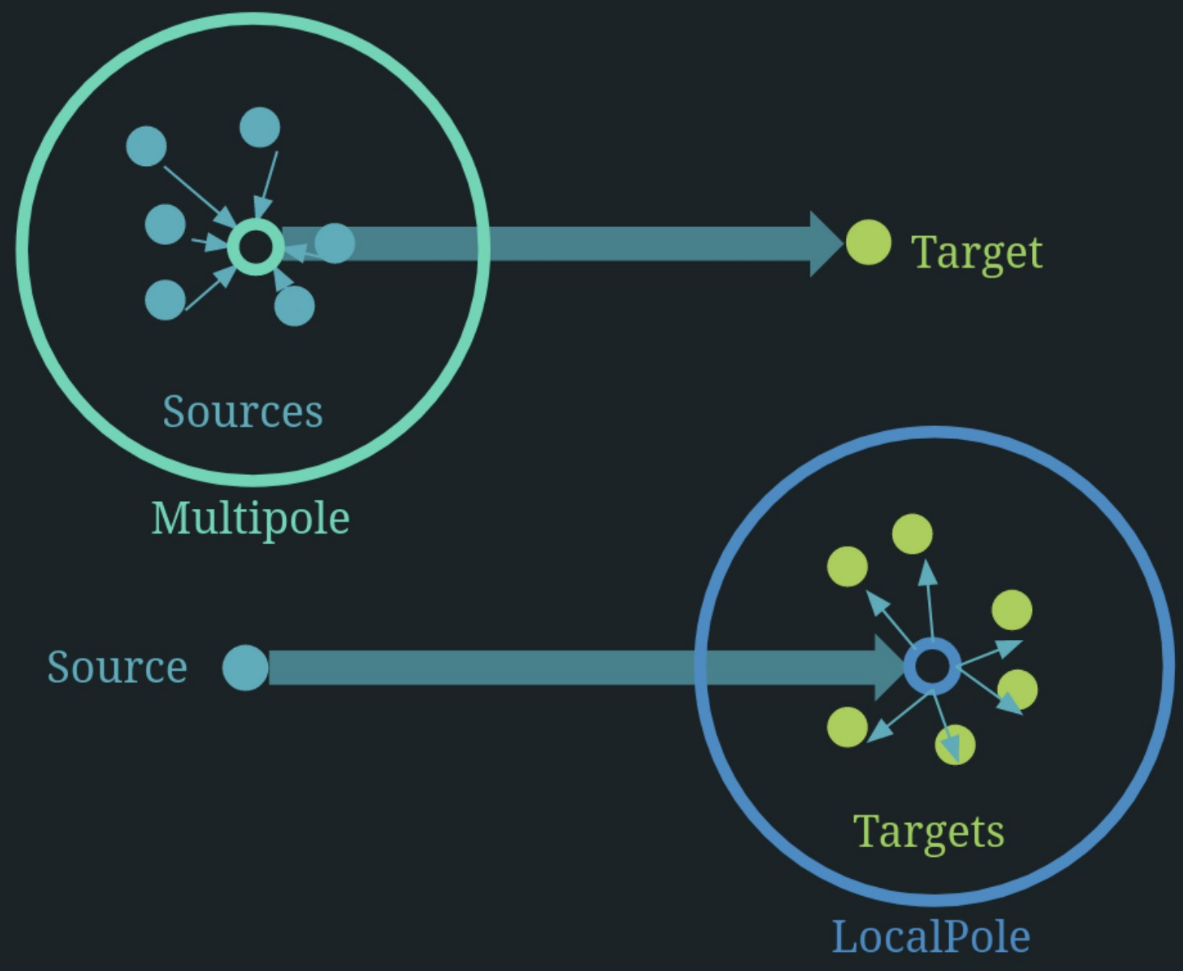
The multipole method uses multipoles and localpoles to represent the field of the sources.

Multipoles represent the field of sources inside a sphere at any target point outside the sphere.

Localpoles represent the field of sources outside a sphere at any target point inside the sphere.

In this sense they are essentially opposites.





[1] L. Greengard and V. Rokhlin. A Fast Algorithm for Particle Simulations. J. Comput. Phys. 73, 325–348 (1987).
[2] J. Kurzak and B. M. Pettitt. Fast multipole methods for particle dynamics. Mol Simul. 2006 ; 32(10-11): 775–790. doi:10.1080/08927020600991161.

Fast Multipole Method (II)

Consider a system with N sources and M targets
Each line represents a field evaluation.

Straight forward Biot-Savart integration leads to complexity O(NXM).

By using the Multipoles and Localpoles as a "middle-man" the complexity is reduced to O(N+M).

This is the essence of the MLFMM.