MIF3: Matlab Interface For Fast Field solvers

*Matlab Interface For Fast Field solvers* is a series of functions and scripts to work with MIT's Fast Field Solvers [[1]](#footnote-1) to generate geometries, automate simulation procedures, and provide extra utilities for planar inductors and wireless power transfer coil designs.

The main objective of this project is to provide a very quick, powerful and accurate tool to estimate the self-inductance, couplings, ac resistance and capacitance between any number of inductors with arbitrary geometries and spatial positions for a huge frequency range. The main limitation of this project is the impossibility of including magnetic materials in the simulations. Accordingly, these tools are specially designed for high frequency coreless magnetics and wireless power transfer systems.

The whole repository can be found here: <https://github.com/JCCopyrights/MIF3>

A general flow diagram of the different functions and data types can be found below:

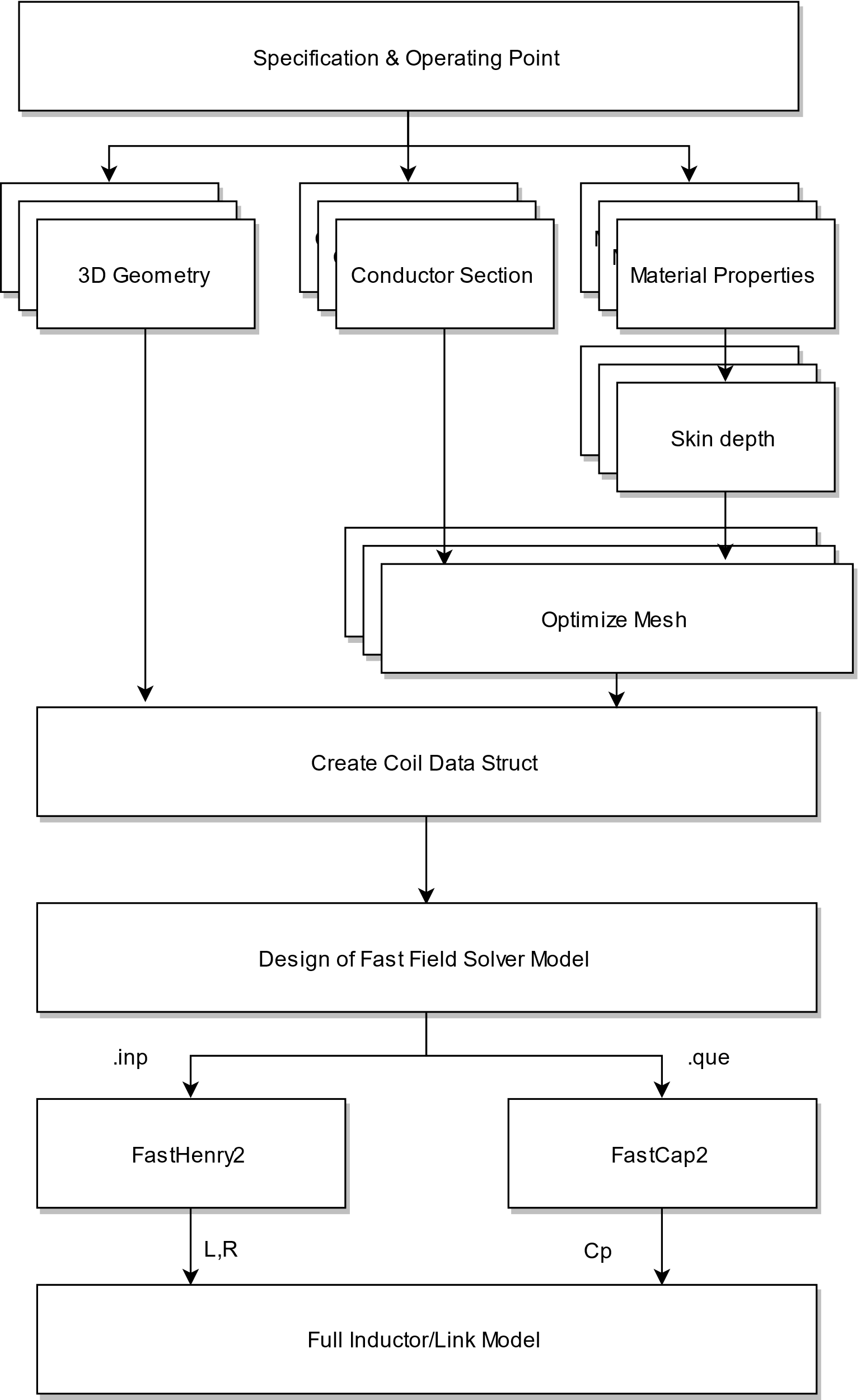


Figure 1: Flow Diagram

The library functions thus, can be classified in three main groups:

A.1. **Geometries:**

Functions that model the geometries of the inductors.

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Function** |
| Solenoid Spiral | Spring with flat turns | solenoid\_spiral |
| Round Spiral | Single layer rounded spiral | round\_spiral |
| Rectangular Spiral | Single layer rectangular spiral | square\_spiral |
| Helix Spiral | Helicoidal spring | helix\_spiral |
| Rectangular Planar Inductor | Multilayer rectangular spiral | rectangular\_planar\_inductor |
| Circular Planar Inductor | Multilayer circular spiral | circular\_planar\_inductor |

Table 1: Geometry Functions

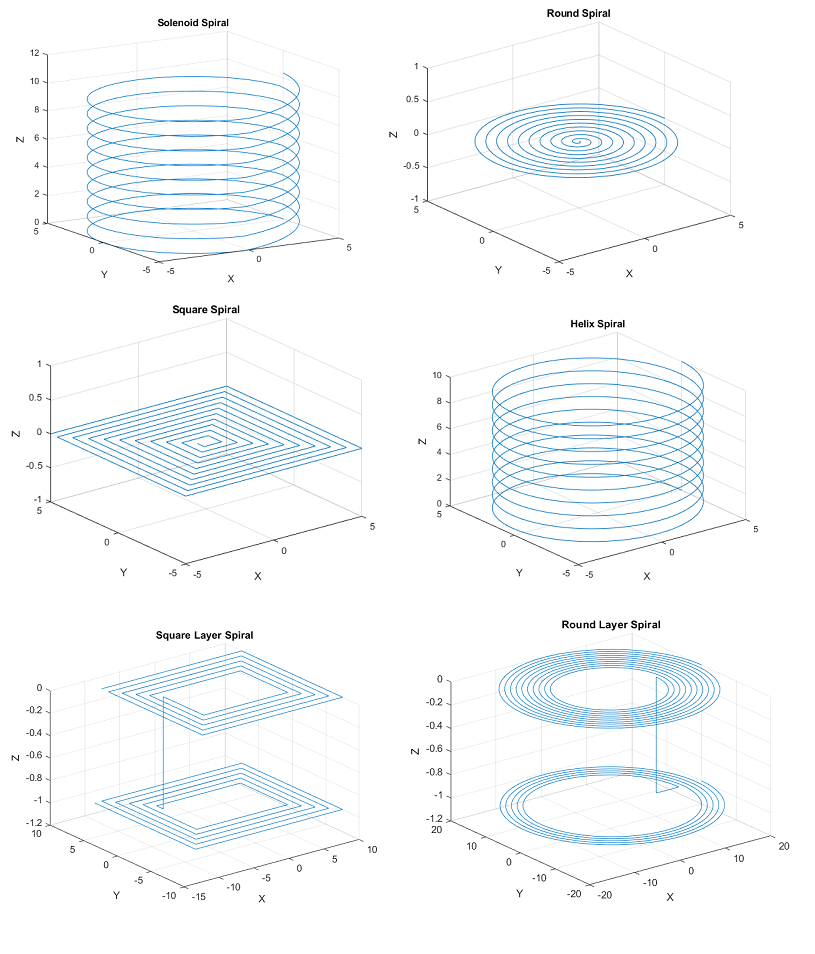


Figure 2: Geometries

A.2. **Fast Field Solver Interface:**

Functions that create the data structures and interfaces to work with Fast Field Solvers

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Function** |
| Generate Coil | Generates a struct with the coil geometry and conductor properties. | generate\_coil |
| Discretization Tools | Calculates a mesh discretization that allows an arbitrary width filament in the edges of the conductors. [\*] | optimize\_discr |
| FastHenry Creator | Creates a. inp model containing all the geometry and conductor data for FastHenry. | fasthenry\_creator |
| FastHenry Runner | Runs the fasthenrymodel and retrieves the results | fasthenry\_runner |
| FastCap Creator | Creates a. list and .que files containing the conductor surfaces | fastcap2\_creator |
| FastCap Runner | Runs the fastcap model and retrieves the Maxwell capacitance matrix | fastcap2\_runner |

Table 2: Fast Field Solver interface functions

[\*] The discretization of the segments is probably the most important characteristic of the model to accurately simulate the inductors. In FastHenry the size of the discretization filaments can be controlled modifying the number of filaments in the width and height of the conductor. In this library the most external filaments (the most critical ones) width and height can be introduced, and the nhinc and nwinc parameters will be calculated in a way that assures the size of these filaments. Every other filament width and height is geometrically increased with a selectable values (rh,wh).

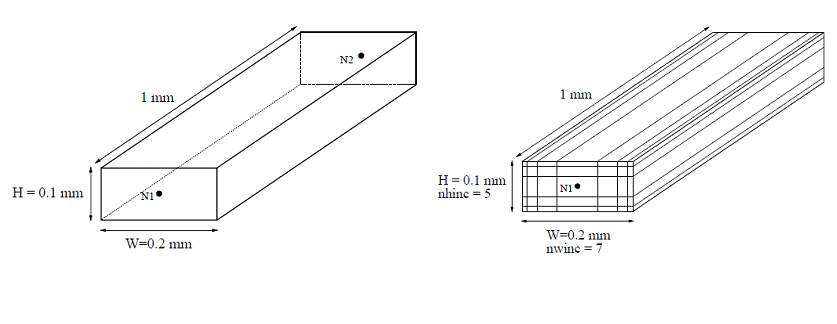


Figure 3: Discretization of conductors

A.3. **Utilities:**

Other functions and utilities that help and complete the functionalities of the main library.

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Function** |
| Import Bode100 | Imports data from a csv file generated with bode100 suite | import\_bode100 |
| Generate Model Bode100 | Takes data imported from bode100 and generates a L, C, R model that matches the data | model\_bode100 |
| Real Coil | Takes parameters from a model and calculates the series impedance equivalents | real\_coil |
| Import raw LTSpice | Takes a .raw file from a Spice simulation and imports it as a struct | LTautomation |
| Run LTSpice | Runs a .asc LTSpice simulation file | LTspice2Matlab |
| Modify LTSpice | Looks for a component in a .asc file and modifies its value | LTmodify |

Table 3: Utilities functions

A.4. **Example Topologies:**

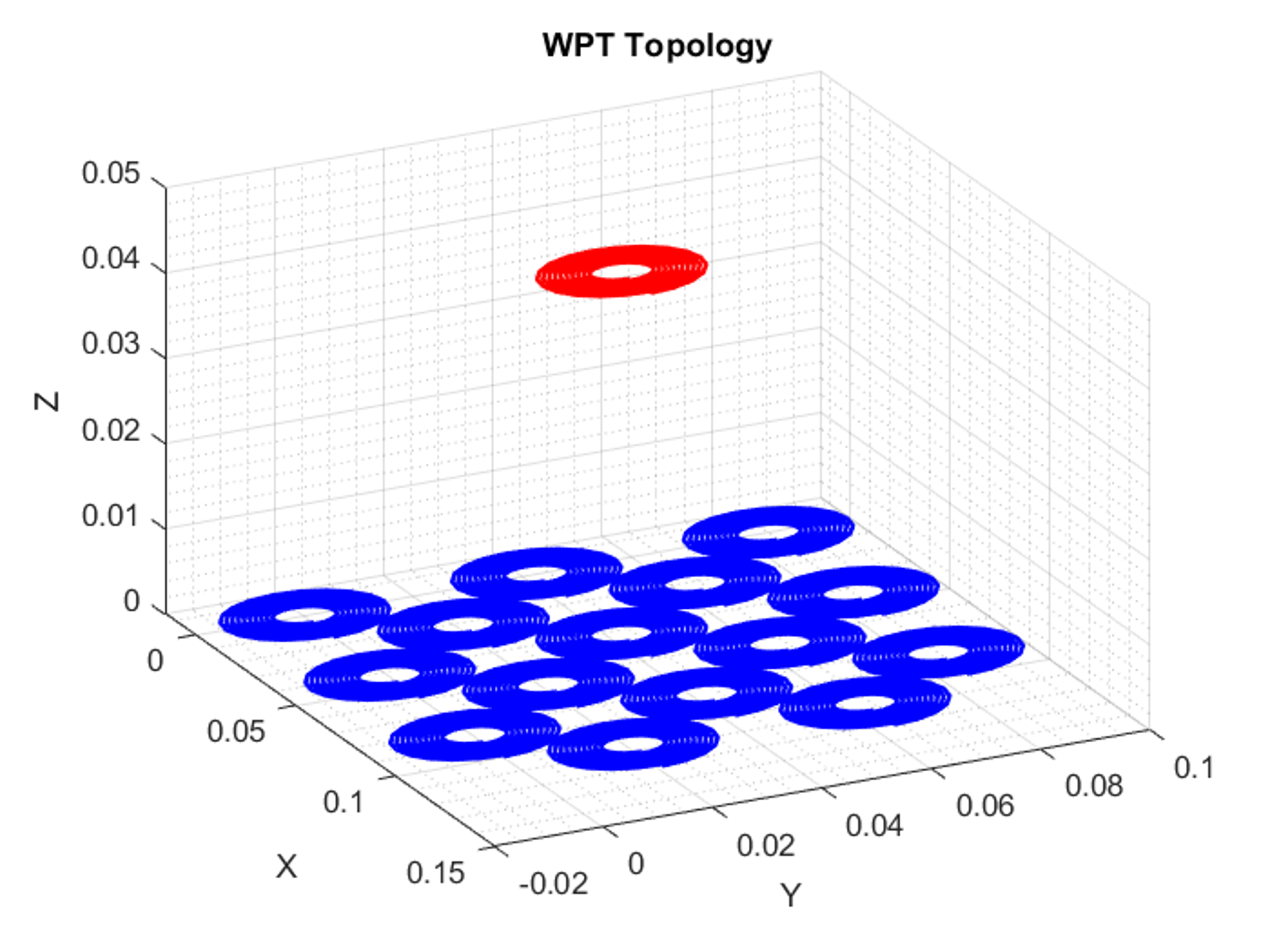
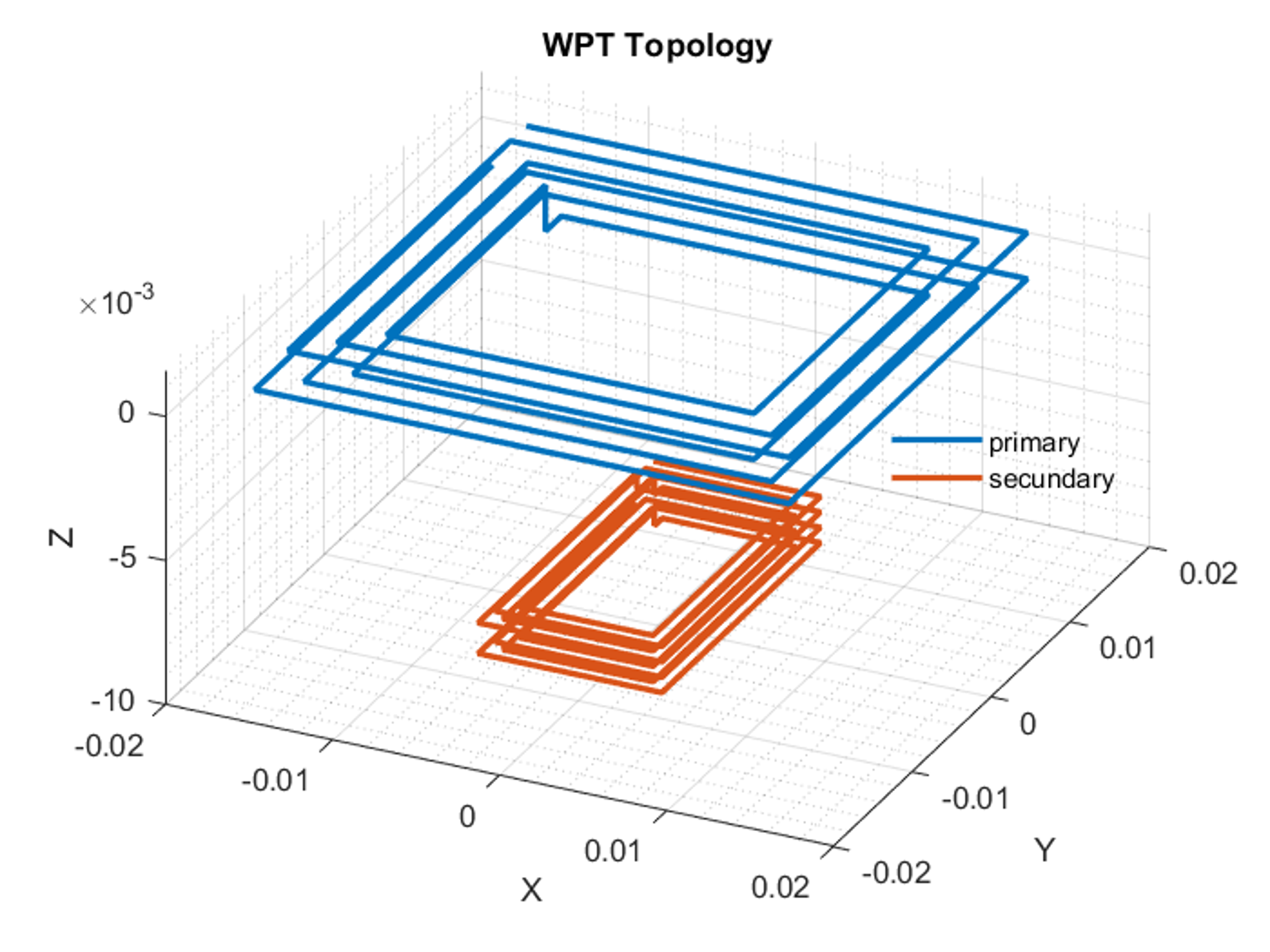
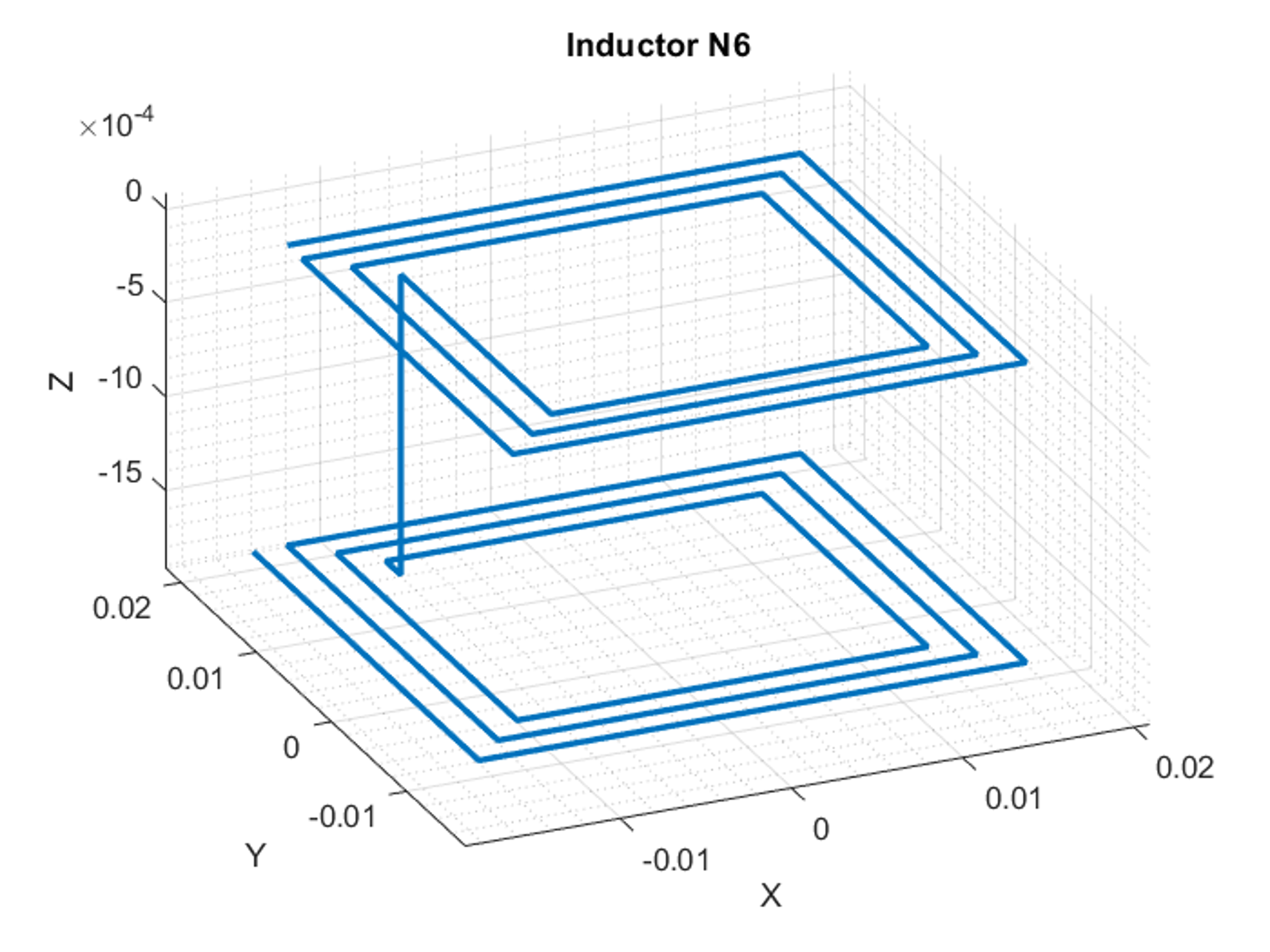


Figure 4: a) Inductor b) WPT system c) Multiple WPT

1. FastFieldSolvers, “White Papers and application notes” [Online], Available <https://www.fastfieldsolvers.com/> [↑](#footnote-ref-1)