A library of MPT Spectral Signature Object Characterisations for Object Identification in Metal Detection

A.A.S. Amad [*1], B. A. Wilson [†1], and P.D. Ledger [‡2]

[1]Zienkiewicz Centre for Computational Engineering, College of Engineering, Swansea University Bay Campus, Swansea, Wales, UK
[2]School of Computing & Mathematics, Keele University, UK

May 21, 2021

Contents

1 Abstract **3**

2 Object - Block **3**
- 2.1 OBJ_Block_mu_1 . . . 3
- 2.2 OBJ_Block_mu_2 . . . 5

3 Object - Cone **5**
- 3.1 OBJ_Cone_NetgenMesh . . . 5
  - 3.1.1 OBJ_Cone_Copper . . . 5
  - 3.1.2 OBJ_Cone_Brass . . . 7
  - 3.1.3 OBJ_Cone_Aluminium . . . 8
- 3.2 OBJ_Cone_StepFileMesh . . . 9
  - 3.2.1 OBJ_Cone_Copper_sig_5.95e7 . . . 9
  - 3.2.2 OBJ_Cone_Copper_sig_5.27e7 . . . 10
  - 3.2.3 OBJ_Cone_Copper_sig_5.8e7 . . . 11
  - 3.2.4 OBJ_Cone_Brass_sig_1.45e7 . . . 12
  - 3.2.5 OBJ_Cone_Brass_sig_1.39e7 . . . 13

4 Object - Cube **13**
- 4.1 OBJ_Cube_mu_1.0001 . . . 13
- 4.2 OBJ_Cube_mu_1.5 . . . 14

5 Object - Disk **16**
- 5.1 OBJ_Disk . . . 16

*a.a.s.amad@swansea.ac.uk
†792104@swansea.ac.uk
‡p.d.ledger@keele.ac.uk

6 Object - Dual Bar **17**
6.1 OBJ_DualBar . . . 17

7 Object - Equivalent Ellipsoids **18**
7.1 OBJ_Sphere . . . 18
7.1.1 OBJ_Sphere_sig_1.5e6 . . . 18
7.1.2 OBJ_Sphere_sig_5.96e6 . . . 19
7.2 OBJ_SpheroidBlock . . . 20
7.2.1 OBJ_SpheroidBlock_mu_1 . . . 20
7.2.2 OBJ_SpheroidBlock_mu_2 . . . 21
7.3 OBJ_SpheroidTetraIrreg . . . 22
7.3.1 OBJ_SpheroidTetraIrreg_mu_1 . . . 22
7.3.2 OBJ_SpheroidTetraIrreg_mu_2 . . . 23
7.4 OBJ_SpheroidTorus . . . 24
7.5 OBJ_SpheroidTwoSpheres . . . 25
7.6 OBJ_SpheroidTwoSpheres . . . 26

8 Object - Genus **27**
8.1 OBJ_TwoTorus . . . 27

9 Object - Keys **28**
9.1 OBJ_Set1 . . . 28
9.1.1 OBJ_Key1_mu_0.0001 . . . 29
9.1.2 OBJ_Key1_mu_1 . . . 30
9.1.3 OBJ_Key2_mu_0.0001 . . . 31
9.1.4 OBJ_Key2_mu_1 . . . 32
9.1.5 OBJ_Key3 . . . 33
9.1.6 OBJ_Key4 . . . 34
9.1.7 OBJ_Key9 . . . 35
9.2 OBJ_Set2 . . . 36
9.2.1 OBJ_Key5_mu_0.0001 . . . 37
9.2.2 OBJ_Key5_mu_1 . . . 38
9.2.3 OBJ_Key6_mu_0.0001 . . . 39
9.2.4 OBJ_Key6_mu_1 . . . 40
9.2.5 OBJ_Key7 . . . 41
9.2.6 OBJ_Key8 . . . 42

10 Object - Tetrahedra **43**
10.1 OBJ_Equilateral_Tetrahedron . . . 43
10.2 OBJ_Irregular_Tetrahedron_mu_1 . . . 45
10.3 OBJ_Irregular_Tetrahedron_mu_2_freq_100 . . . 46
10.4 OBJ_Irregular_Tetrahedron_mu_2_freq_81 . . . 47

11 Object - Torus **48**
11.1 OBJ_Torus . . . 48

12 Object - Two Overlapping Spheres **49**
12.1 OBJ_TwoSpheres . . . 49

1 Abstract

This documentation describes the geometries and settings used in the MPT-Calculator program for a range of different shapes (`https://github.com/BAWilson94/MPT-Calculator`) in order to allow their MPT spectral signature to be computed. In each case a description of the geometry is provided together with the settings for the MPT calculator. Details of the algorithm used to produce these simulations can be found in [1].

2 Object - Block

2.1 OBJ_Block_mu_1

The geometry is a block $B_\alpha := \alpha B$. The non-dimensional rectangular block B has dimensions $[-0.5, -1, -1.5] \times [0.5, 1, 1.5]$. The scaling parameter is $\alpha = 0.01\,\mathrm{m}$. The material properties are $\mu_r = 1$ and $\sigma_* = 1 \times 10^7$ S/m. The object B and the region around it to a truncating boundary in the form of the box $[-100, 100]^3$ is discretised by $35\,915$ unstructured tetrahedral elements. The resulting MPT has three independent coefficients $(\mathcal{M})_{11}$, $(\mathcal{M})_{22}$, and $(\mathcal{M})_{33}$ at each frequency. The discretised object B is presented in Figure 7 and the simulation is using a polynomial order of $p = 3$.

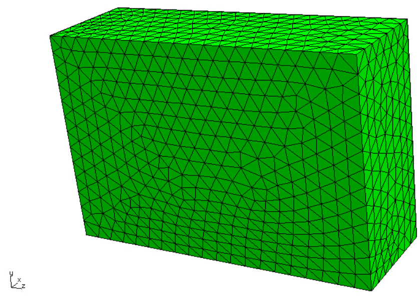


Figure 1: OBJ_Block_mu_1: Discretised object.

The parameter settings to reproduce the example is presented in Table 1.

main.py

Geometry = "block.geo"	alpha = 0.01	MeshSize = 3
Order = 3	Start = 2	Finish = 9
Points = 100	Single = False	Omega = 100000
Pod = True	MultiProcessing = True	

Settings.py

CPUs = 4	BigProblem = False	PODPoints = 13
PODTol = 1e-08	OldMesh = False	PlotPod = True
PODErrorBars = False	EddyCurrentTest = False	vtk_output = False
Refine_vtk = False	FolderName = "Default"	Solver = "bddc"
epsi = 1e-12	Maxsteps = 5000	Tolerance = 1e-07
	ngsglobals.msg_level = 3	

Table 1: OBJ_Block_mu_1: data of simulation.

2.2 OBJ_Block_mu_2

The geometry is a block $B_\alpha := \alpha B$, the configuration and the mesh are the same as presented in Section 2.1. The material properties are $\mu_r = 2$ and $\sigma_* = 1 \times 10^7$ S/m. The parameter settings to reproduce the example is presented in Table 2.

main.py

`Geometry = "block.geo"`	`alpha = 0.01`	`MeshSize = 3`
`Order = 3, 4`	`Start = 0`	`Finish = 8`
`Points = 100`	`Single = False`	`Omega = 100000`
`Pod = True`	`MultiProcessing = True`	

Settings.py

`CPUs = 8`	`BigProblem = False`	`PODPoints = 13`
`PODTol = 1e-08`	`OldMesh = False`	`PlotPod = True`
`PODErrorBars = False`	`EddyCurrentTest = False`	`vtk_output = False`
`Refine_vtk = False`	`FolderName = "Default"`	`Solver = "bddc"`
`epsi = 1e-12`	`Maxsteps = 1500`	`Tolerance = 1e-08`
	`ngsglobals.msg_level = 3`	

Table 2: OBJ_Block_mu_2: data of simulation.

3 Object - Cone

3.1 OBJ_Cone_NetgenMesh

3.1.1 OBJ_Cone_Copper

The geometry is a cone $B_\alpha := \alpha B$. The non-dimensional cone B has its base in the $x_1 - x_2$ plane with radius 15, the height of the cone is 30 and the tip is cut-off using a radius of 1. The scaling parameter is $\alpha = 0.001\,\mathrm{m}$. The material properties are $\mu_r = 0.99991$ and $\sigma_* = 5.95 \times 10^7$ S/m. The discretisation uses an object primitive in Netgen and its edges are not rounded. The object B and the region around it to a truncating boundary in the form of the box $[-1000, 1000]^3$ is discretised by $114\,098$ unstructured tetrahedral elements. The resulting MPT has two independent coefficients $(\mathcal{M})_{11} = (\mathcal{M})_{22}$ and $(\mathcal{M})_{33}$ at each frequency. The discretised object B is presented in Figure 2 and the simulation is using a polynomial order $p = 4$.
The parameter settings to reproduce the example is presented in Table 3.

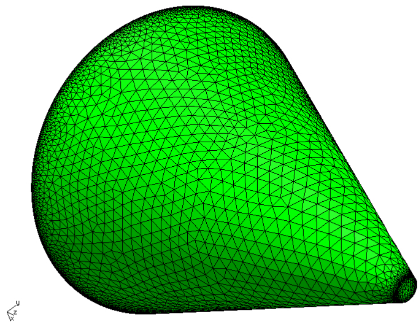


Figure 2: Cone Copper: Discretised object.

main.py

Geometry = "cone.geo"	alpha = 0.001	MeshSize = 1
Order = 4	Start = 1	Finish = 6
Points = 71	Single = False	Omega = 100000
Pod = False	MultiProcessing = True	

Settings.py

CPUs = 4	BigProblem = True	PODPoints = 13
PODTol = 1e-08	OldMesh = True	PlotPod = True
PODErrorBars = False	EddyCurrentTest = False	vtk_output = False
Refine_vtk = False	FolderName = "Default"	Solver = "bddc"
epsi = 1e-09	Maxsteps = 5000	Tolerance = 1e-08
	ngsglobals.msg_level = 3	

Table 3: Cone Cooper: data of simulation.

3.1.2 OBJ_Cone_Brass

The geometry is a cone $B_\alpha := \alpha B$. The non-dimensional cone B has its base in the $x_1 - x_2$ plane with radius 1.5, the height of the cone is 3 and the tip is cut-off using a radius of 0.01. The scaling parameter is $\alpha = 0.01\,\mathrm{m}$. The material properties are $\mu_r = 1$ and $\sigma_* = 1.5 \times 10^7$ S/m. The discretisation uses an object primitive in Netgen and its edges are not rounded. The object B and the region around it to a truncating boundary in the form of a sphere with radius 200 is discretised by $26\,289$ unstructured tetrahedral elements. The resulting MPT has two independent coefficients $(\mathcal{M})_{11} = (\mathcal{M})_{22}$ and $(\mathcal{M})_{33}$ at each frequency. The discretised object B is presented in Figure 3 and the simulation is using a polynomial order $p = 3$.

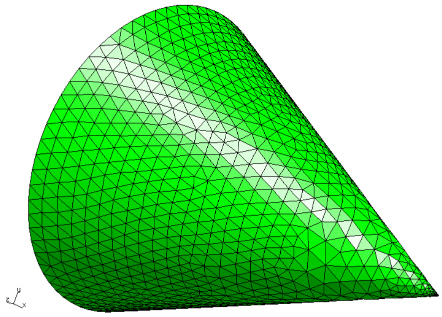


Figure 3: Cone Brass: Discretised object.

The parameter settings to reproduce the example is presented in Table 4.

`main.py`

`Geometry = "cone_Brass.geo"`	`alpha = 0.01`	`MeshSize = 1`
`Order = 3`	`Start = 0`	`Finish = 9`
`Points = 100`	`Single = False`	`Omega = 100000`
`Pod = True`	`MultiProcessing = True`	

`Settings.py`

`CPUs = 16`	`BigProblem = True`	`PODPoints = 13`
`PODTol = 1e-08`	`OldMesh = False`	`PlotPod = True`
`PODErrorBars = False`	`EddyCurrentTest = False`	`vtk_output = False`
`Refine_vtk = False`	`FolderName = "Default"`	`Solver = "bddc"`
`epsi = 1e-12`	`Maxsteps = 5000`	`Tolerance = 1e-10`
	`ngsglobals.msg_level = 3`	

Table 4: Cone Brass: data of simulation.

3.1.3 OBJ_Cone_Aluminium

The geometry is a cone $B_\alpha := \alpha B$. The non-dimensional cone B has its base in the $x_1 - x_2$ plane with radius 1.5, the height of the cone is 3 and the tip is cut-off using a radius of 0.005. The geometry is as described in Section 3.1.2. The scaling parameter is $\alpha = 0.01\,\mathrm{m}$. The material properties are $\mu_r = 1.000021$ and $\sigma_* = 3.8 \times 10^7$ S/m. The discretisation uses an object primitive in Netgen and its edges are not rounded. The object B and the region around it to a truncating boundary in the form of a sphere with radius 200 is discretised by 25 798 unstructured tetrahedral elements. The resulting MPT has two independent coefficients $(\mathcal{M})_{11} = (\mathcal{M})_{22}$ and $(\mathcal{M})_{33}$ at each frequency. The discretised object B is presented in Figure 3 and the simulation is using a polynomial order $p = 3$. The parameter settings to reproduce the example is presented in Table 5.

main.py

Geometry = "cone_Aluminium.geo"	alpha = 0.01	MeshSize = 1
Order = 3	Start = 0	Finish = 9
Points = 100	Single = False	Omega = 100000
Pod = True	MultiProcessing = True	

Settings.py

CPUs = 16	BigProblem = True	PODPoints = 13
PODTol = 1e-08	OldMesh = False	PlotPod = True
PODErrorBars = False	EddyCurrentTest = False	vtk_output = False
Refine_vtk = False	FolderName = "Default"	Solver = "bddc"
epsi = 1e-09	Maxsteps = 1500	Tolerance = 1e-08
	ngsglobals.msg_level = 3	

Table 5: Cone Aluminium: data of simulation.

3.2 OBJ_Cone_StepFileMesh

3.2.1 OBJ_Cone_Copper_sig_5.95e7

The geometry is a cone $B_\alpha := \alpha B$. The non-dimensional cone B has its base in the $x_1 - x_2$ plane with radius 15, the height of the cone is 30 and the tip is cut-off using a radius of 1. The scaling parameter is $\alpha = 0.001\,\mathrm{m}$. The material properties are $\mu_r = 0.99991$ and $\sigma_* = 5.95 \times 10^7$ S/m. The object geometry is obtained from step file, but the discretisation uses Netgen. The object B and the region around it to a truncating boundary in the form of the box $[-1000, 1000]^3$ is discretised by 122 012 unstructured tetrahedral elements. The resulting MPT has two independent coefficients $(\mathcal{M})_{11} = (\mathcal{M})_{22}$ and $(\mathcal{M})_{33}$ at each frequency. The discretised object B is presented in Figure 4 and the simulation is using polynomial orders $p = 3, 4$.

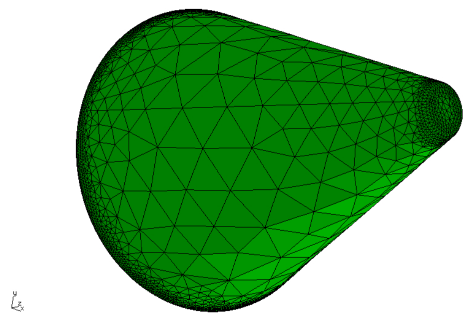


Figure 4: OBJ_Cone_Copper_sig_5.95e7: Discretised object.

The parameter settings to reproduce the example is presented in Table 6.

`main.py`

`Geometry = "coneJohn.geo"`	`alpha = 0.001`	`MeshSize = 1`
`Order = 3`	`Start = 2`	`Finish = 8`
`Points = 71`	`Single = False`	`Omega = 100000`
`Pod = True`	`MultiProcessing = False`	

`Settings.py`

`CPUs = 10`	`BigProblem = True`	`PODPoints = 13`
`PODTol = 1e-08`	`OldMesh = True`	`PlotPod = True`
`PODErrorBars = True`	`EddyCurrentTest = True`	`vtk_output = False`
`Refine_vtk = False`	`FolderName = "Default"`	`Solver = "bddc"`
`epsi = 1e-09`	`Maxsteps = 5000`	`Tolerance = 1e-08`
	`ngsglobals.msg_level = 3`	

Table 6: OBJ_Cone_Copper_sig_5.95e7: data of simulation.

3.2.2 OBJ_Cone_Copper_sig_5.27e7

The geometry is a cone $B_\alpha := \alpha B$, the configuration and the mesh are the same as presented in Section 3.2.1. The material properties are $\mu_r = 0.99991$ and $\sigma_* = 5.27 \times 10^7$ S/m. The parameter settings to reproduce the example is presented in Table 7.

`main.py`

`Geometry = "ConeCopperExp.geo"`	`alpha = 0.001`	`MeshSize = 1`
`Order = 4`	`Start = 1`	`Finish = 8`
`Points = 71`	`Single = False`	`Omega = 100000`
`Pod = True`	`MultiProcessing = True`	

`Settings.py`

`CPUs = 10`	`BigProblem = True`	`PODPoints = 13`
`PODTol = 1e-08`	`OldMesh = True`	`PlotPod = True`
`PODErrorBars = False`	`EddyCurrentTest = False`	`vtk_output = False`
`Refine_vtk = False`	`FolderName = "Default"`	`Solver = "bddc"`
`epsi = 1e-09`	`Maxsteps = 5000`	`Tolerance = 1e-08`
	`ngsglobals.msg_level = 3`	

Table 7: OBJ_Cone_Copper_sig_5.27e7: data of simulation.

3.2.3 OBJ_Cone_Copper_sig_5.8e7

The geometry is a cone $B_\alpha := \alpha B$, the configuration and the mesh are the same as presented in Section 3.2.1. The material properties are $\mu_r = 0.99991$ and $\sigma_* = 5.8 \times 10^7$ S/m. The parameter settings to reproduce the example is presented in Table 8.

main.py

`Geometry = "coneJohn.geo"`	`alpha = 0.001`	`MeshSize = 1`
`Order = 3`	`Start = 2`	`Finish = 8`
`Points = 71`	`Single = False`	`Omega = 100000`
`Pod = True`	`MultiProcessing = False`	

Settings.py

`CPUs = 10`	`BigProblem = False`	`PODPoints = 13`
`PODTol = 1e-08`	`OldMesh = True`	`PlotPod = True`
`PODErrorBars = True`	`EddyCurrentTest = True`	`vtk_output = False`
`Refine_vtk = False`	`FolderName = "Default"`	`Solver = "bddc"`
`epsi = 1e-09`	`Maxsteps = 5000`	`Tolerance = 1e-08`
	`ngsglobals.msg_level = 3`	

Table 8: OBJ_Cone_Copper_sig_5.8e7: data of simulation.

3.2.4 OBJ_Cone_Brass_sig_1.45e7

The geometry is a cone $B_\alpha := \alpha B$, the configuration and the mesh are the same as presented in Section 3.2.1. The material properties are $\mu_r = 1$ and $\sigma_* = 1.45 \times 10^7$ S/m. The parameter settings to reproduce the example is presented in Table 9, for the case $p = 3$ the `PODPoints` is 28, while for $p = 4$ the `PODPoints` is 13.

`main.py`

`Geometry = "ConeBrass.geo"`	`alpha = 0.001`	`MeshSize = 1`
`Order = 3,4`	`Start = 1`	`Finish = 8`
`Points = 71`	`Single = False`	`Omega = 100000`
`Pod = True`	`MultiProcessing = True`	

`Settings.py`

`CPUs = 8`	`BigProblem = True`	`PODPoints = 28, 13`
`PODTol = 1e-08`	`OldMesh = True`	`PlotPod = True`
`PODErrorBars = False`	`EddyCurrentTest = False`	`vtk_output = False`
`Refine_vtk = False`	`FolderName = "Default"`	`Solver = "bddc"`
`epsi = 1e-09`	`Maxsteps = 5000`	`Tolerance = 1e-08`
	`ngsglobals.msg_level = 3`	

Table 9: OBJ_Cone_Brass_sig_1.45e7: data of simulation.

3.2.5 OBJ_Cone_Brass_sig_1.39e7

The geometry is a cone $B_\alpha := \alpha B$, the configuration and the mesh are the same as presented in Section 3.2.1. The material properties are $\mu_r = 1$ and $\sigma_* = 1.45 \times 10^7$ S/m. The parameter settings to reproduce the example is presented in Table 10, for the case $p = 3$ the `PODPoints` is 28, while for $p = 4$ the `PODPoints` is 13.

`main.py`

`Geometry = "ConeBrassExp.geo"`	`alpha = 0.001`	`MeshSize = 1`
`Order = 4`	`Start = 1`	`Finish = 8`
`Points = 71`	`Single = False`	`Omega = 100000`
`Pod = True`	`MultiProcessing = True`	

`Settings.py`

`CPUs = 8`	`BigProblem = True`	`PODPoints = 13`
`PODTol = 1e-08`	`OldMesh = True`	`PlotPod = True`
`PODErrorBars = False`	`EddyCurrentTest = False`	`vtk_output = False`
`Refine_vtk = False`	`FolderName = "Default"`	`Solver = "bddc"`
`epsi = 1e-09`	`Maxsteps = 5000`	`Tolerance = 1e-08`
	`ngsglobals.msg_level = 3`	

Table 10: OBJ_Cone_Brass_sig_1.39e7: data of simulation.

4 Object - Cube

4.1 OBJ_Cube_mu_1.0001

The geometry is a cube $B_\alpha := \alpha B$, B is a unit cube $[-0.5, 0.5]^3$ centred at the origin. The scaling parameter is $\alpha = 0.01\,\mathrm{m}$. The material properties are $\mu_r = 1.0001$ and $\sigma_* = 2 \times 10^6$ S/m. The object B and the region around it to a truncating boundary in the form of the box $[-100, 100]^3$ is discretised by 32 210 unstructured tetrahedral elements. The resulting MPT has one independent coefficient $(\mathcal{M})_{11} = (\mathcal{M})_{22} = (\mathcal{M})_{33}$ at each frequency. The discretised object B is presented in Figure 5 and the simulation is using a polynomial order $p = 4$.
The parameter settings to reproduce the example is presented in Table 11.

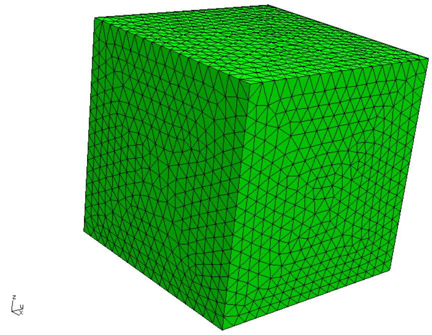


Figure 5: OBJ_Cube_mu_1.0001: Object discretised.

main.py

`Geometry = "cube.geo"`	`alpha = 0.01`	`MeshSize = 1`
`Order = 4`	`Start = 1`	`Finish = 6`
`Points = 71`	`Single = False`	`Omega = 100000`
`Pod = False`	`MultiProcessing = True`	

Settings.py

`CPUs = 8`	`BigProblem = True`	`PODPoints = 13`
`PODTol = 1e-08`	`OldMesh = False`	`PlotPod = True`
`PODErrorBars = False`	`EddyCurrentTest = True`	`vtk_output = False`
`Refine_vtk = False`	`FolderName = "Default"`	`Solver = "bddc"`
`epsi = 1e-09`	`Maxsteps = 5000`	`Tolerance = 1e-08`
	`ngsglobals.msg_level = 3`	

Table 11: OBJ_Cube_mu_1.0001: data of simulation.

4.2 OBJ_Cube_mu_1.5

The geometry is a cube $B_\alpha := \alpha B$, B is a unit cube $[-0.5, 0.5]^3$ centred at the origin. The scaling parameter is $\alpha = 0.01\,\text{m}$. The material properties are $\mu_r = 1.5$ and $\sigma_* = 1.5 \times 10^6$ S/m. The object B and the region around it to a truncating boundary in the form of the box $[-100, 100]^3$ is discretised by 36 919 unstructured tetrahedral elements. The resulting MPT has one independent coefficient $(\mathcal{M})_{11} = (\mathcal{M})_{22} = (\mathcal{M})_{33}$ at each frequency. The discretised object B is presented in Figure 6 and the simulation is using polynomial orders $p = 3, 4$.
The parameter settings to reproduce the example is presented in Table 12.

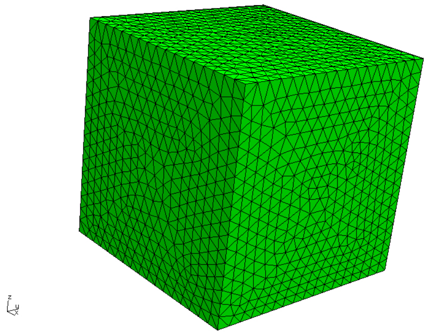


Figure 6: OBJ_Cube_mu_1.5: Object discretised.

main.py

Geometry = "cube.geo"	alpha = 0.01	MeshSize = 3
Order = 3,4	Start = 0	Finish = 8
Points = 100	Single = False	Omega = 100000
Pod = True	MultiProcessing = True	

Settings.py

CPUs = 10	BigProblem = False	PODPoints = 13
PODTol = 1e-04	OldMesh = False	PlotPod = True
PODErrorBars = False	EddyCurrentTest = True	vtk_output = False
Refine_vtk = False	FolderName = "Default"	Solver = "local"
epsi = 1e-12	Maxsteps = 5000	Tolerance = 1e-10
	ngsglobals.msg_level = 3	

Table 12: OBJ_Cube_mu_1.5: data of simulation.

5 Object - Disk

5.1 OBJ_Disk

The geometry is a disk $B_\alpha := \alpha B$. The non-dimensional disk B has a top and bottom in the shape of a circle with radius 1.53 and height 0.22. The scaling parameter is $\alpha = 0.01\,\mathrm{m}$. The material properties are $\mu_r = 1$ and $\sigma_* = 5.8 \times 10^7$ S/m. The object B and the region around it to a truncating boundary in the form of the box $[-100, 100]^3$ is discretised by $63\,645$ unstructured tetrahedral elements. The resulting MPT has two independent coefficients $(\mathcal{M})_{11} = (\mathcal{M})_{22}$, and $(\mathcal{M})_{33}$ at each frequency. The discretised object B is presented in Figure 7 and the simulation is using polynomial orders $p = 3, 4$.

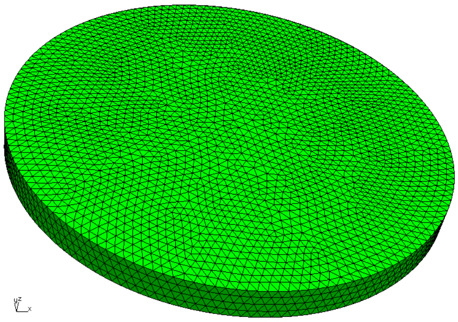


Figure 7: OBJ_Disk: Discretised object.

The parameter settings to reproduce the example is presented in Table 13.

`main.py`

`Geometry = "disk.geo"`	`alpha = 0.01`	`MeshSize = 1`
`Order = 4`	`Start = 1`	`Finish = 8`
`Points = 71`	`Single = False`	`Omega = 100000`
`Pod = False`	`MultiProcessing = True`	

`Settings.py`

`CPUs = 10`	`BigProblem = True`	`PODPoints = 13`
`PODTol = 1e-08`	`OldMesh = False`	`PlotPod = True`
`PODErrorBars = False`	`EddyCurrentTest = True`	`vtk_output = False`
`Refine_vtk = False`	`FolderName = "Default"`	`Solver = "bddc"`
`epsi = 1e-09`	`Maxsteps = 5000`	`Tolerance = 1e-08`
	`ngsglobals.msg_level = 3`	

Table 13: OBJ_Disk: data of simulation.

6 Object - Dual Bar

6.1 OBJ_DualBar

The geometry is a dual Bar $B_\alpha := \alpha B$. The non-dimensional rectangular block is made up of two cubes joined together $B = B_1 \bigcup B_2$, with $B_1 := [-1,0,0] \times [0,1,1]$ and $B_2 := [0,0,0] \times [1,1,1]$. The scaling parameter is $\alpha = 0.01\,\mathrm{m}$. The material properties are $\mu_r^{(1)} = 1$, $\sigma_*^{(1)} = 1 \times 10^6$ S/m for B_1 and $\mu_r^{(2)} = 1$, $\sigma_*^{(2)} = 1 \times 10^8$ S/m for B_2. The object B and the region around it to a truncating boundary in the form of a sphere with radius 100 is discretised by 30 209 unstructured tetrahedral elements. The resulting MPT has three independent coefficients $(\mathcal{M})_{11}$, $(\mathcal{M})_{22}$, and $(\mathcal{M})_{33}$ at each frequency. The discretised object B is presented in Figure 8 and the simulation is using polynomial order $p = 3$.

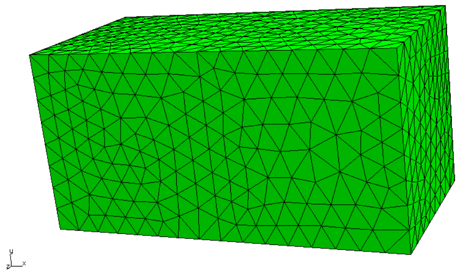


Figure 8: OBJ_DualBar: Discretised object.

The parameter settings to reproduce the example is presented in Table 14.

`main.py`

`Geometry = "DualBar.geo"`	`alpha = 0.01`	`MeshSize = 3`
`Order = 3`	`Start = 2`	`Finish = 8`
`Points = 81`	`Single = False`	`Omega = 100000`
`Pod = False`	`MultiProcessing = True`	

`Settings.py`

`CPUs = 5`	`BigProblem = False`	`PODPoints = 23`
`PODTol = 1e-04`	`OldMesh = False`	`PlotPod = True`
`PODErrorBars = True`	`EddyCurrentTest = True`	`vtk_output = False`
`Refine_vtk = False`	`FolderName = "Default"`	`Solver = "bddc"`
`epsi = 1e-12`	`Maxsteps = 1500`	`Tolerance = 1e-10`
	`ngsglobals.msg_level = 3`	

Table 14: OBJ_DualBar: data of simulation.

7 Object - Equivalent Ellipsoids

In [2, Section 3], there is a detailed explanation of how to obtain equivalent ellipsoids.

7.1 OBJ_Sphere

7.1.1 OBJ_Sphere_sig_1.5e6

The geometry is a sphere $B_\alpha := \alpha B$. The non-dimensional sphere B is a sphere with radius 0.6213. The scaling parameter is $\alpha = 0.01\,\mathrm{m}$. The material properties are $\mu_r = 1.5$ and $\sigma_* = 1.5 \times 10^6$ S/m. The object B and the region around it to a truncating boundary in the form of a sphere with radius 200 is discretised by 23 649 unstructured tetrahedral elements. The resulting MPT has one independent coefficient $(\mathcal{M})_{11} = (\mathcal{M})_{22} = (\mathcal{M})_{33}$ at each frequency. The simulation is using polynomial order $p = 3$ and parameter settings to reproduce the example is presented in Table 15.

main.py

Geometry = "sphere.geo"	alpha = 0.01	MeshSize = 3
Order = 3	Start = 0	Finish = 8
Points = 100	Single = False	Omega = 100000
Pod = True	MultiProcessing = True	

Settings.py

CPUs = 10	BigProblem = True	PODPoints = 20
PODTol = 1e-04	OldMesh = False	PlotPod = True
PODErrorBars = False	EddyCurrentTest = True	vtk_output = False
Refine_vtk = False	FolderName = "Default"	Solver = "bddc"
epsi = 1e-12	Maxsteps = 5000	Tolerance = 1e-10
	ngsglobals.msg_level = 3	

Table 15: OBJ_Sphere_sig_1.5e6: data of simulation.

7.1.2 OBJ_Sphere_sig_5.96e6

The geometry is a sphere $B_\alpha := \alpha B$. The non-dimensional sphere B is a unitary sphere. The scaling parameter is $\alpha = 0.01$ m. The material properties are $\mu_r = 1.5$ and $\sigma_* = 5.96 \times 10^6$ S/m. The object B and the region around it to a truncating boundary in the form of a sphere with radius 200 is discretised by 26 751 unstructured tetrahedral elements. The resulting MPT has one independent coefficient $(\mathcal{M})_{11} = (\mathcal{M})_{22} = (\mathcal{M})_{33}$ at each frequency. The simulation is using polynomial order $p = 3$ and parameter settings to reproduce the example is presented in Table 16.

`main.py`

`Geometry = "sphere.geo"`	`alpha = 0.01`	`MeshSize = 2`
`Order = 3`	`Start = 2`	`Finish = 8`
`Points = 81`	`Single = False`	`Omega = 100000`
`Pod = True`	`MultiProcessing = True`	

`Settings.py`

`CPUs = 5`	`BigProblem = True`	`PODPoints = 21`
`PODTol = 1e-04`	`OldMesh = True`	`PlotPod = True`
`PODErrorBars = True`	`EddyCurrentTest = True`	`vtk_output = False`
`Refine_vtk = False`	`FolderName = "Default"`	`Solver = "bddc"`
`epsi = 1e-12`	`Maxsteps = 1500`	`Tolerance = 1e-10`
	`ngsglobals.msg_level = 3`	

Table 16: OBJ_Sphere_sig_5.96e6: data of simulation.

7.2 OBJ_SpheroidBlock

7.2.1 OBJ_SpheroidBlock_mu_1

The geometry is an ellipsoid $B_\alpha := \alpha B$. The non-dimensional ellipsoid B has radii 0.6877, 1.2366, and 1.7047. The material properties are $\mu_r = 1$ and $\sigma_* = 1 \times 10^7$ S/m. The object B and the region around it to a truncating boundary in the form of a sphere with radius 100 is discretised by $35\,651$ unstructured tetrahedral elements. The resulting MPT has three independent coefficients $(\mathcal{M})_{11}$, $(\mathcal{M})_{22}$, and $(\mathcal{M})_{33}$ at each frequency. The simulation is using polynomial order $p = 3$ and parameter settings to reproduce the example is presented in Table 17.

`main.py`

`Geometry = "spheroidBlock.geo"`	`alpha = 0.01`	`MeshSize = 3`
`Order = 3`	`Start = 2`	`Finish = 9`
`Points = 100`	`Single = False`	`Omega = 100000`
`Pod = True`	`MultiProcessing = True`	

`Settings.py`

`CPUs = 6`	`BigProblem = False`	`PODPoints = 13`
`PODTol = 1e-08`	`OldMesh = False`	`PlotPod = True`
`PODErrorBars = False`	`EddyCurrentTest = True`	`vtk_output = False`
`Refine_vtk = False`	`FolderName = "Default"`	`Solver = "bddc"`
`epsi = 1e-12`	`Maxsteps = 5000`	`Tolerance = 1e-07`
	`ngsglobals.msg_level = 3`	

Table 17: OBJ_SpheroidBlock_mu_1: data of simulation.

7.2.2 OBJ_SpheroidBlock_mu_2

The geometry is an ellipsoid $B_\alpha := \alpha B$. The non-dimensional ellipsoid B has radii 0.6877, 1.2366, and 1.7047. The material properties are $\mu_r = 1$ and $\sigma_* = 1 \times 10^7$ S/m. The object B and the region around it to a truncating boundary in the form of a sphere with radius 100 is discretised by 35 651 unstructured tetrahedral elements. The resulting MPT has three independent coefficients $(\mathcal{M})_{11}$, $(\mathcal{M})_{22}$, and $(\mathcal{M})_{33}$ at each frequency. The simulation is using polynomial order $p = 3$ and parameter settings to reproduce the example is presented in Table 18.

`main.py`

`Geometry = "spheroidBlock.geo"`	`alpha = 0.01`	`MeshSize = 3`
`Order = 3, 4`	`Start = 0`	`Finish = 8`
`Points = 100`	`Single = False`	`Omega = 100000`
`Pod = True`	`MultiProcessing = True`	

`Settings.py`

`CPUs = 8`	`BigProblem = False`	`PODPoints = 13`
`PODTol = 1e-08`	`OldMesh = False`	`PlotPod = True`
`PODErrorBars = False`	`EddyCurrentTest = True`	`vtk_output = False`
`Refine_vtk = False`	`FolderName = "Default"`	`Solver = "bddc"`
`epsi = 1e-12`	`Maxsteps = 1500`	`Tolerance = 1e-08`
	`ngsglobals.msg_level = 3`	

Table 18: OBJ_SpheroidBlock_mu_2: data of simulation.

7.3 OBJ_SpheroidTetraIrreg

7.3.1 OBJ_SpheroidTetraIrreg_mu_1

The geometry is an ellipsoid $B_\alpha := \alpha B$. The non-dimensional ellipsoid B has radii 1.3693, 1.9090, and 2.9404. The material properties are $\mu_r = 1$ and $\sigma_* = 1 \times 10^7$ S/m. The object B and the region around it to a truncating boundary in the form of a sphere with radius 100 is discretised by 35 651 unstructured tetrahedral elements. The resulting MPT has three independent coefficients $(\mathcal{M})_{11}$, $(\mathcal{M})_{22}$, and $(\mathcal{M})_{33}$ at each frequency. The simulation is using polynomial order $p = 3$ and parameter settings to reproduce the example is presented in Table 19.

main.py

Geometry = "spheroidTetraIrreg.geo"	alpha = 0.01	MeshSize = 3
Order = 3	Start = 2	Finish = 9
Points = 100	Single = False	Omega = 100000
Pod = True	MultiProcessing = True	

Settings.py

CPUs = 6	BigProblem = False	PODPoints = 13
PODTol = 1e-08	OldMesh = False	PlotPod = True
PODErrorBars = False	EddyCurrentTest = True	vtk_output = False
Refine_vtk = False	FolderName = "Default"	Solver = "bddc"
epsi = 1e-12	Maxsteps = 5000	Tolerance = 1e-07
	ngsglobals.msg_level = 3	

Table 19: OBJ_SpheroidTetraIrreg_mu_1: data of simulation.

7.3.2 OBJ_SpheroidTetraIrreg_mu_2

The geometry is an ellipsoid $B_\alpha := \alpha B$. The non-dimensional ellipsoid B has radii 1.4426, 1.8797, and 2.4243. The material properties are $\mu_r = 2$ and $\sigma_* = 5.96 \times 10^6$ S/m. The object B and the region around it to a truncating boundary in the form of a sphere with radius 100 is discretised by 27 970 unstructured tetrahedral elements. The resulting MPT has three independent coefficients $(\mathcal{M})_{11}$, $(\mathcal{M})_{22}$, and $(\mathcal{M})_{33}$ at each frequency. The simulation is using polynomial order $p = 3$ and parameter settings to reproduce the example is presented in Table 20.

`main.py`

`Geometry = "spheroidTetraIrreg.geo"`	`alpha = 0.01`	`MeshSize = 3`
`Order = 3`	`Start = 0`	`Finish = 8`
`Points = 100`	`Single = False`	`Omega = 100000`
`Pod = True`	`MultiProcessing = True`	

`Settings.py`

`CPUs = 6`	`BigProblem = False`	`PODPoints = 13`
`PODTol = 1e-08`	`OldMesh = False`	`PlotPod = True`
`PODErrorBars = False`	`EddyCurrentTest = True`	`vtk_output = False`
`Refine_vtk = False`	`FolderName = "Default"`	`Solver = "bddc"`
`epsi = 1e-12`	`Maxsteps = 5000`	`Tolerance = 1e-07`
	`ngsglobals.msg_level = 3`	

Table 20: OBJ_SpheroidTetraIrreg_mu_2: data of simulation.

7.4 OBJ_SpheroidTorus

The geometry is an ellipsoid $B_\alpha := \alpha B$. The non-dimensional ellipsoid B has radii 1.4426, 1.8797, and 2.4243. The material properties are $\mu_r = 1.5$ and $\sigma_* = 5 \times 10^5$ S/m. The object B and the region around it to a truncating boundary in the form of a sphere with radius 100 is discretised by 36 469 unstructured tetrahedral elements. The resulting MPT has three independent coefficients $(\mathcal{M})_{11}$, $(\mathcal{M})_{22}$, and $(\mathcal{M})_{33}$ at each frequency. The simulation is using polynomial order $p = 3$ and parameter settings to reproduce the example is presented in Table 21.

`main.py`

`Geometry = "spheroidTetraIrreg.geo"`	`alpha = 0.01`	`MeshSize = 3`
`Order = 3, 4`	`Start = 0`	`Finish = 8`
`Points = 100`	`Single = False`	`Omega = 100000`
`Pod = True`	`MultiProcessing = True`	

`Settings.py`

`CPUs = 6`	`BigProblem = False`	`PODPoints = 13`
`PODTol = 1e-08`	`OldMesh = False`	`PlotPod = True`
`PODErrorBars = False`	`EddyCurrentTest = True`	`vtk_output = False`
`Refine_vtk = False`	`FolderName = "Default"`	`Solver = "bddc"`
`epsi = 1e-12`	`Maxsteps = 5000`	`Tolerance = 1e-07`
	`ngsglobals.msg_level = 3`	

Table 21: OBJ_SpheroidTorus: data of simulation.

7.5 OBJ_SpheroidTwoSpheres

The geometry is a spheroid $B_\alpha := \alpha B$. The non-dimensional spheroid B has radii 1.3639, 2.6332. The material properties are $\mu_r = 1$ and $\sigma_* = 1 \times 10^7$ S/m. The object B and the region around it to a truncating boundary in the form of a sphere with radius 100 is discretised by 28 159 unstructured tetrahedral elements. The resulting MPT has two independent coefficients $(\mathcal{M})_{11} = (\mathcal{M})_{22}$, and $(\mathcal{M})_{33}$ at each frequency. The simulation is using polynomial order $p = 3$ and parameter settings to reproduce the example is presented in Table 22.

`main.py`

`Geometry = "spheroidTetraIrreg.geo"`	`alpha = 0.01`	`MeshSize = 3`
`Order = 3, 4`	`Start = 0`	`Finish = 8`
`Points = 100`	`Single = False`	`Omega = 100000`
`Pod = True`	`MultiProcessing = True`	

`Settings.py`

`CPUs = 8`	`BigProblem = False`	`PODPoints = 13`
`PODTol = 1e-08`	`OldMesh = False`	`PlotPod = True`
`PODErrorBars = False`	`EddyCurrentTest = True`	`vtk_output = False`
`Refine_vtk = False`	`FolderName = "Default"`	`Solver = "bddc"`
`epsi = 1e-12`	`Maxsteps = 1500`	`Tolerance = 1e-08`
	`ngsglobals.msg_level = 3`	

Table 22: OBJ_SpheroidTwoSpheres: data of simulation.

7.6 OBJ_SpheroidTwoSpheres

The geometry is an ellipsoid $B_\alpha := \alpha B$. The non-dimensional ellipsoid B has radii 1.1698, 3.3291, and 5.3693. The material properties are $\mu_r = 1$ and $\sigma_* = 1 \times 10^7$ S/m. The object B and the region around it to a truncating boundary in the form of a sphere with radius 100 is discretised by 32 888 unstructured tetrahedral elements. The resulting MPT has three independent coefficients $(\mathcal{M})_{11}$, $(\mathcal{M})_{22}$, and $(\mathcal{M})_{33}$ at each frequency. The simulation is using polynomial order $p = 3$ and parameter settings to reproduce the example is presented in Table 23.

`main.py`

`Geometry = "spheroidTwoTorus.geo"`	`alpha = 0.01`	`MeshSize = 1`
`Order = 3`	`Start = 2`	`Finish = 9`
`Points = 100`	`Single = False`	`Omega = 100000`
`Pod = True`	`MultiProcessing = True`	

`Settings.py`

`CPUs = 8`	`BigProblem = False`	`PODPoints = 13`
`PODTol = 1e-08`	`OldMesh = False`	`PlotPod = True`
`PODErrorBars = False`	`EddyCurrentTest = True`	`vtk_output = False`
`Refine_vtk = False`	`FolderName = "Default"`	`Solver = "bddc"`
`epsi = 1e-12`	`Maxsteps = 5000`	`Tolerance = 1e-10`
	`ngsglobals.msg_level = 3`	

Table 23: OBJ_SpheroidTwoTorus: data of simulation.

8 Object - Genus

8.1 OBJ_TwoTorus

The geometry is a genus $B_\alpha := \alpha B$. The non-dimensional genus $B = B_1 \bigcup B_2$ is constructed using two overlapping torii, such as B_1 is centred at $(-2.5, 0, 0)$ and B_2 is centred at $(2.5, 0, 0)$, both torus major and minor radii, 2 and 1, respectively. The scaling parameter is $\alpha = 0.01\,\mathrm{m}$. The material properties are $\mu_r = 1$ and $\sigma_* = 1 \times 10^7$ S/m. The object B and the region around it to a truncating boundary in the form of a sphere with radius 100 is discretised by $48\,100$ unstructured tetrahedral elements. The resulting MPT has three independent coefficients $(\mathcal{M})_{11}$, $(\mathcal{M})_{22}$, and $(\mathcal{M})_{33}$ at each frequency. The discretised object B is presented in Figure 9 and the simulation is using polynomial order $p = 3$.

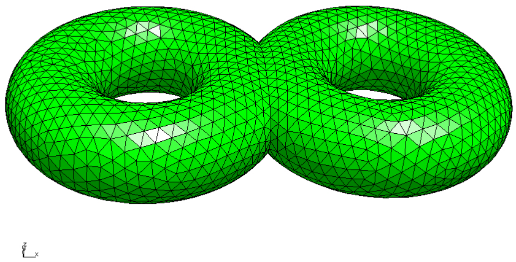


Figure 9: OBJ_TwoTorus: Discretised object.

The parameter settings to reproduce the example is presented in Table 24.

main.py

Geometry = "twoTorus.geo"	alpha = 0.01	MeshSize = 2
Order = 3	Start = 2	Finish = 9
Points = 100	Single = False	Omega = 100000
Pod = True	MultiProcessing = True	

Settings.py

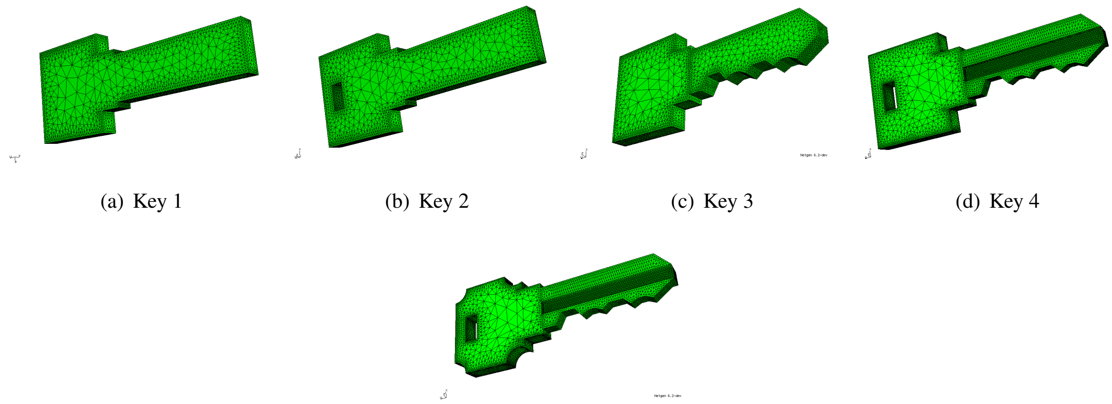
CPUs = 8	BigProblem = False	PODPoints = 13
PODTol = 1e-08	OldMesh = False	PlotPod = True
PODErrorBars = False	EddyCurrentTest = False	vtk_output = False
Refine_vtk = False	FolderName = "Default"	Solver = "bddc"
epsi = 1e-12	Maxsteps = 5000	Tolerance = 1e-10
	ngsglobals.msg_level = 3	

Table 24: OBJ_TwoTorus: data of simulation.

9 Object - Keys

9.1 OBJ_Set1

In Figure 10 is presented the set of the discretised keys. In key 1 and 2 there are three independent coefficients. The key 3 has four independent coefficients and keys 4 and 9 have six independent coefficients, for more details see [2, Section 5].



(a) Key 1 (b) Key 2 (c) Key 3 (d) Key 4

(e) Key 9

Figure 10: Set 1 keys: Discretised objects.

9.1.1 OBJ_Key1_mu_0.0001

The geometry is the key 1 $B_\alpha := \alpha B$. The scaling parameter is $\alpha = 0.001\,\mathrm{m}$. The material properties are $\mu_r = 0.0001$ and $\sigma_* = 1.5 \times 10^7$ S/m. The object B and the region around it to a truncating boundary in the form of the box $[-1000, 1000]^3$ is discretised by $56\,241$ unstructured tetrahedral elements. The parameter settings to reproduce the example is presented in Table 25.

`main.py`

`Geometry = "key_type1.geo"`	`alpha = 0.001`	`MeshSize = 1`
`Order = 3`	`Start = 2`	`Finish = 8`
`Points = 100`	`Single = False`	`Omega = 1000000`
`Pod = True`	`MultiProcessing = True`	

`Settings.py`

`CPUs = 4`	`BigProblem = False`	`PODPoints = 13`
`PODTol = 1e-08`	`OldMesh = True`	`PlotPod = True`
`PODErrorBars = True`	`EddyCurrentTest = True`	`vtk_output = False`
`Refine_vtk = True`	`FolderName = "Default"`	`Solver = "bddc"`
`epsi = 1e-09`	`Maxsteps = 5000`	`Tolerance = 1e-07`
	`ngsglobals.msg_level = 0`	

Table 25: OBJ_Key1_mu_0.0001: data of simulation.

9.1.2 OBJ_Key1_mu_1

The geometry is the key 1 $B_\alpha := \alpha B$. The scaling parameter is $\alpha = 0.001\,\mathrm{m}$. The material properties are $\mu_r = 1$ and $\sigma_* = 1.5 \times 10^7$ S/m. The object B and the region around it to a truncating boundary in the form of the box $[-1000, 1000]^3$ is discretised by $56\,241$ unstructured tetrahedral elements. The parameter settings to reproduce the example is presented in Table 26, for $p = 0, 1, 2, 3, 4$.

main.py

`Geometry = "key_type1.geo"`	`alpha = 0.001`	`MeshSize = 1`
`Order = 4`	`Start = 2`	`Finish = 8`
`Points = 100`	`Single = False`	`Omega = 1000000`
`Pod = True`	`MultiProcessing = True`	

Settings.py

`CPUs = 8`	`BigProblem = True`	`PODPoints = 31`
`PODTol = 1e-08`	`OldMesh = True`	`PlotPod = True`
`PODErrorBars = True`	`EddyCurrentTest = False`	`vtk_output = False`
`Refine_vtk = True`	`FolderName = "Default"`	`Solver = "bddc"`
`epsi = 1e-09`	`Maxsteps = 5000`	`Tolerance = 1e-07`
	`ngsglobals.msg_level = 0`	

Table 26: OBJ_Key1_mu_1: data of simulation.

9.1.3 OBJ_Key2_mu_0.0001

The geometry is the key 2 $B_\alpha := \alpha B$. The scaling parameter is $\alpha = 0.001\,\mathrm{m}$. The material properties are $\mu_r = 0.0001$ and $\sigma_* = 1.5 \times 10^7$ S/m. The object B and the region around it to a truncating boundary in the form of the box $[-1000, 1000]^3$ is discretised by $66\,535$ unstructured tetrahedral elements. The parameter settings to reproduce the example is presented in Table 27.

main.py

`Geometry = "key_type2.geo"`	`alpha = 0.001`	`MeshSize = 1`
`Order = 3`	`Start = 2`	`Finish = 8`
`Points = 100`	`Single = False`	`Omega = 1000000`
`Pod = True`	`MultiProcessing = True`	

Settings.py

`CPUs = 4`	`BigProblem = False`	`PODPoints = 13`
`PODTol = 1e-08`	`OldMesh = True`	`PlotPod = True`
`PODErrorBars = True`	`EddyCurrentTest = True`	`vtk_output = False`
`Refine_vtk = True`	`FolderName = "Default"`	`Solver = "bddc"`
`epsi = 1e-09`	`Maxsteps = 5000`	`Tolerance = 1e-07`
	`ngsglobals.msg_level = 0`	

Table 27: OBJ_Key2_mu_0.0001: data of simulation.

9.1.4 OBJ_Key2_mu_1

The geometry is the key 2 $B_\alpha := \alpha B$. The scaling parameter is $\alpha = 0.001\,\mathrm{m}$. The material properties are $\mu_r = 1$ and $\sigma_* = 1.5 \times 10^7$ S/m. The object B and the region around it to a truncating boundary in the form of the box $[-1000, 1000]^3$ is discretised by 66 535 unstructured tetrahedral elements. The parameter settings to reproduce the example is presented in Table 28, for $p = 0, 1, 2, 3$.

main.py

Geometry = "key_type2.geo"	alpha = 0.001	MeshSize = 1
Order = 4	Start = 2	Finish = 8
Points = 100	Single = False	Omega = 1000000
Pod = True	MultiProcessing = True	

Settings.py

CPUs = 8	BigProblem = True	PODPoints = 13
PODTol = 1e-08	OldMesh = True	PlotPod = True
PODErrorBars = True	EddyCurrentTest = False	vtk_output = False
Refine_vtk = True	FolderName = "Default"	Solver = "bddc"
epsi = 1e-09	Maxsteps = 5000	Tolerance = 1e-08
	ngsglobals.msg_level = 0	

Table 28: OBJ_Key2_mu_1: data of simulation.

9.1.5 OBJ_Key3

The geometry is the key 3 $B_\alpha := \alpha B$. The scaling parameter is $\alpha = 0.001\,\mathrm{m}$. The material properties are $\mu_r = 1$ and $\sigma_* = 1.5 \times 10^7$ S/m. The object B and the region around it to a truncating boundary in the form of the box $[-1000, 1000]^3$ is discretised by 73 795 unstructured tetrahedral elements. The parameter settings to reproduce the example is presented in Table 29, for $p = 0, 1, 2, 3, 4, 5$.

main.py

`Geometry = "key_type3.geo"`	`alpha = 0.001`	`MeshSize = 1`
`Order = 4`	`Start = 2`	`Finish = 8`
`Points = 100`	`Single = False`	`Omega = 1000000`
`Pod = True`	`MultiProcessing = True`	

Settings.py

`CPUs = 8`	`BigProblem = True`	`PODPoints = 21`
`PODTol = 1e-08`	`OldMesh = True`	`PlotPod = True`
`PODErrorBars = True`	`EddyCurrentTest = False`	`vtk_output = False`
`Refine_vtk = True`	`FolderName = "Default"`	`Solver = "bddc"`
`epsi = 1e-09`	`Maxsteps = 5000`	`Tolerance = 1e-08`
	`ngsglobals.msg_level = 0`	

Table 29: OBJ_Key3: data of simulation.

9.1.6 OBJ_Key4

The geometry is the key 4 $B_\alpha := \alpha B$. The scaling parameter is $\alpha = 0.001\,\mathrm{m}$. The material properties are $\mu_r = 1$ and $\sigma_* = 1.5 \times 10^7$ S/m. The object B and the region around it to a truncating boundary in the form of the box $[-1000, 1000]^3$ is discretised by 108 523 unstructured tetrahedral elements. The parameter settings to reproduce the example is presented in Table 30, for $p = 0, 1, 2, 3, 4$.

`main.py`

`Geometry = "key_type4.geo"`	`alpha = 0.001`	`MeshSize = 1`
`Order = 4`	`Start = 2`	`Finish = 8`
`Points = 100`	`Single = False`	`Omega = 1000000`
`Pod = True`	`MultiProcessing = True`	

`Settings.py`

`CPUs = 8`	`BigProblem = True`	`PODPoints = 21`
`PODTol = 1e-08`	`OldMesh = True`	`PlotPod = True`
`PODErrorBars = True`	`EddyCurrentTest = False`	`vtk_output = False`
`Refine_vtk = True`	`FolderName = "Default"`	`Solver = "bddc"`
`epsi = 1e-09`	`Maxsteps = 5000`	`Tolerance = 1e-08`
	`ngsglobals.msg_level = 0`	

Table 30: OBJ_Key4: data of simulation.

9.1.7 OBJ_Key9

The geometry is the key 9 $B_\alpha := \alpha B$. The scaling parameter is $\alpha = 0.001\,\mathrm{m}$. The material properties are $\mu_r = 1$ and $\sigma_* = 1.5 \times 10^7$ S/m. The object B and the region around it to a truncating boundary in the form of the box $[-1000, 1000]^3$ is discretised by 107 106 unstructured tetrahedral elements. The parameter settings to reproduce the example is presented in Table 31, for $p = 0, 1, 2, 3, 4$.

main.py

`Geometry = "key_type9.geo"`	`alpha = 0.001`	`MeshSize = 1`
`Order = 4`	`Start = 2`	`Finish = 8`
`Points = 100`	`Single = False`	`Omega = 1000000`
`Pod = True`	`MultiProcessing = True`	

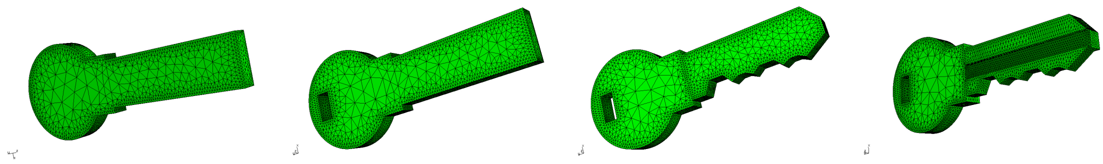
Settings.py

`CPUs = 8`	`BigProblem = True`	`PODPoints = 13`
`PODTol = 1e-08`	`OldMesh = True`	`PlotPod = True`
`PODErrorBars = True`	`EddyCurrentTest = False`	`vtk_output = False`
`Refine_vtk = True`	`FolderName = "Default"`	`Solver = "bddc"`
`epsi = 1e-09`	`Maxsteps = 5000`	`Tolerance = 1e-08`
	`ngsglobals.msg_level = 0`	

Table 31: OBJ_Key9: data of simulation.

9.2 OBJ_Set2

In Figure 11 is presented the set of the discretised keys. In key 5 and 6 there are three independent coefficients. The key 7 has four independent coefficients and keys 8 has six independent coefficients, for more details see [2, Section 5].



(a) Key 5 (b) Key 6 (c) Key 7 (d) Key 8

Figure 11: Set 2 keys: Discretised objects.

9.2.1 OBJ_Key5_mu_0.0001

The geometry is the key 5 $B_\alpha := \alpha B$. The scaling parameter is $\alpha = 0.001\,\mathrm{m}$. The material properties are $\mu_r = 0.0001$ and $\sigma_* = 1.5 \times 10^7$ S/m. The object B and the region around it to a truncating boundary in the form of the box $[-1000, 1000]^3$ is discretised by 51 726 unstructured tetrahedral elements. The parameter settings to reproduce the example is presented in Table 32.

main.py

`Geometry = "key_type5.geo"`	`alpha = 0.001`	`MeshSize = 1`
`Order = 3`	`Start = 2`	`Finish = 8`
`Points = 100`	`Single = False`	`Omega = 1000000`
`Pod = True`	`MultiProcessing = True`	

Settings.py

`CPUs = 4`	`BigProblem = False`	`PODPoints = 13`
`PODTol = 1e-08`	`OldMesh = True`	`PlotPod = True`
`PODErrorBars = True`	`EddyCurrentTest = True`	`vtk_output = False`
`Refine_vtk = True`	`FolderName = "Default"`	`Solver = "bddc"`
`epsi = 1e-09`	`Maxsteps = 5000`	`Tolerance = 1e-07`
	`ngsglobals.msg_level = 0`	

Table 32: OBJ_Key5_mu_0.0001: data of simulation.

9.2.2 OBJ_Key5_mu_1

The geometry is the key 5 $B_\alpha := \alpha B$. The scaling parameter is $\alpha = 0.001\,\mathrm{m}$. The material properties are $\mu_r = 1$ and $\sigma_* = 1.5 \times 10^7$ S/m. The object B and the region around it to a truncating boundary in the form of the box $[-1000, 1000]^3$ is discretised by 51 726 unstructured tetrahedral elements. The parameter settings to reproduce the example is presented in Table 33, for $p = 0, 1, 2, 3$.

main.py

Geometry = "key_type5.geo"	alpha = 0.001	MeshSize = 1
Order = 4	Start = 2	Finish = 8
Points = 100	Single = False	Omega = 1000000
Pod = True	MultiProcessing = True	

Settings.py

CPUs = 8	BigProblem = True	PODPoints = 13
PODTol = 1e-08	OldMesh = True	PlotPod = True
PODErrorBars = True	EddyCurrentTest = False	vtk_output = False
Refine_vtk = True	FolderName = "Default"	Solver = "bddc"
epsi = 1e-09	Maxsteps = 5000	Tolerance = 1e-07
	ngsglobals.msg_level = 0	

Table 33: OBJ_Key5_mu_1: data of simulation.

9.2.3 OBJ_Key6_mu_0.0001

The geometry is the key 6 $B_\alpha := \alpha B$. The scaling parameter is $\alpha = 0.001\,\mathrm{m}$. The material properties are $\mu_r = 0.0001$ and $\sigma_* = 1.5 \times 10^7$ S/m. The object B and the region around it to a truncating boundary in the form of the box $[-1000, 1000]^3$ is discretised by $57\,024$ unstructured tetrahedral elements. The parameter settings to reproduce the example is presented in Table 34.

main.py

Geometry = "key_type6.geo"	alpha = 0.001	MeshSize = 1
Order = 3	Start = 2	Finish = 8
Points = 100	Single = False	Omega = 1000000
Pod = True	MultiProcessing = True	

Settings.py

CPUs = 4	BigProblem = False	PODPoints = 13
PODTol = 1e-08	OldMesh = True	PlotPod = True
PODErrorBars = True	EddyCurrentTest = True	vtk_output = False
Refine_vtk = True	FolderName = "Default"	Solver = "bddc"
epsi = 1e-09	Maxsteps = 5000	Tolerance = 1e-07
	ngsglobals.msg_level = 0	

Table 34: OBJ_Key6_mu_0.0001: data of simulation.

9.2.4 OBJ_Key6_mu_1

The geometry is the key 6 $B_\alpha := \alpha B$. The scaling parameter is $\alpha = 0.001\,\mathrm{m}$. The material properties are $\mu_r = 1$ and $\sigma_* = 1.5 \times 10^7$ S/m. The object B and the region around it to a truncating boundary in the form of the box $[-1000, 1000]^3$ is discretised by $57\,024$ unstructured tetrahedral elements. The parameter settings to reproduce the example is presented in Table 35, for $p = 0, 1, 2, 3$.

main.py

`Geometry = "key_type6.geo"`	`alpha = 0.001`	`MeshSize = 1`
`Order = 4`	`Start = 2`	`Finish = 8`
`Points = 100`	`Single = False`	`Omega = 1000000`
`Pod = True`	`MultiProcessing = True`	

Settings.py

`CPUs = 8`	`BigProblem = True`	`PODPoints = 13`
`PODTol = 1e-08`	`OldMesh = True`	`PlotPod = True`
`PODErrorBars = True`	`EddyCurrentTest = False`	`vtk_output = False`
`Refine_vtk = True`	`FolderName = "Default"`	`Solver = "bddc"`
`epsi = 1e-09`	`Maxsteps = 5000`	`Tolerance = 1e-07`
	`ngsglobals.msg_level = 0`	

Table 35: OBJ_Key6_mu_1: data of simulation.

9.2.5 OBJ_Key7

The geometry is the key 7 $B_\alpha := \alpha B$. The scaling parameter is $\alpha = 0.001\,\mathrm{m}$. The material properties are $\mu_r = 1$ and $\sigma_* = 1.5 \times 10^7$ S/m. The object B and the region around it to a truncating boundary in the form of the box $[-1000, 1000]^3$ is discretised by 69 316 unstructured tetrahedral elements. The parameter settings to reproduce the example is presented in Table 36, for $p = 0, 1, 2, 3$.

`main.py`

`Geometry = "key_type7.geo"`	`alpha = 0.001`	`MeshSize = 1`
`Order = 4`	`Start = 2`	`Finish = 8`
`Points = 100`	`Single = False`	`Omega = 1000000`
`Pod = True`	`MultiProcessing = True`	

`Settings.py`

`CPUs = 8`	`BigProblem = True`	`PODPoints = 13`
`PODTol = 1e-08`	`OldMesh = True`	`PlotPod = True`
`PODErrorBars = True`	`EddyCurrentTest = False`	`vtk_output = False`
`Refine_vtk = True`	`FolderName = "Default"`	`Solver = "bddc"`
`epsi = 1e-09`	`Maxsteps = 5000`	`Tolerance = 1e-08`
	`ngsglobals.msg_level = 0`	

Table 36: OBJ_Key7: data of simulation.

9.2.6 OBJ_Key8

The geometry is the key 8 $B_\alpha := \alpha B$. The scaling parameter is $\alpha = 0.001\,\mathrm{m}$. The material properties are $\mu_r = 1$ and $\sigma_* = 1.5 \times 10^7$ S/m. The object B and the region around it to a truncating boundary in the form of the box $[-1000, 1000]^3$ is discretised by $84\,421$ unstructured tetrahedral elements. The parameter settings to reproduce the example is presented in Table 37, for $p = 0, 1, 2, 3$.

main.py

`Geometry = "key_type8.geo"`	`alpha = 0.001`	`MeshSize = 1`
`Order = 4`	`Start = 2`	`Finish = 8`
`Points = 100`	`Single = False`	`Omega = 1000000`
`Pod = True`	`MultiProcessing = True`	

Settings.py

`CPUs = 8`	`BigProblem = True`	`PODPoints = 13`
`PODTol = 1e-08`	`OldMesh = True`	`PlotPod = True`
`PODErrorBars = True`	`EddyCurrentTest = False`	`vtk_output = False`
`Refine_vtk = True`	`FolderName = "Default"`	`Solver = "bddc"`
`epsi = 1e-09`	`Maxsteps = 5000`	`Tolerance = 1e-08`
	`ngsglobals.msg_level = 0`	

Table 37: OBJ_Key8: data of simulation.

10 Object - Tetrahedra

10.1 OBJ_Equilateral_Tetrahedron

The geometry is a tetrahedron $B_\alpha := \alpha B$. The non-dimensional irregular tetrahedron B is chosen to have vertices at the locations

$$\boldsymbol{v}_1 = \begin{pmatrix} -1 \\ 1 \\ -1 \end{pmatrix}, \qquad \boldsymbol{v}_2 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \qquad \boldsymbol{v}_3 = \begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix}, \qquad \boldsymbol{v}_4 = \begin{pmatrix} 1 \\ -1 \\ -1 \end{pmatrix}. \tag{10.1}$$

The scaling parameter is $\alpha = 0.01\,\mathrm{m}$. The material properties are $\mu_r = 1$ and $\sigma_* = 1 \times 10^7$ S/m. The object B and the region around it to a truncating boundary in the form of the box $[-100, 100]^3$ is discretised by $19\,446$ unstructured tetrahedral elements. The resulting MPT has one independent coefficient $(\mathcal{M})_{11} = (\mathcal{M})_{22} = (\mathcal{M})_{33}$ at each frequency. The discretised object B is presented in Figure 7 and the simulation is using polynomial order $p = 3$.

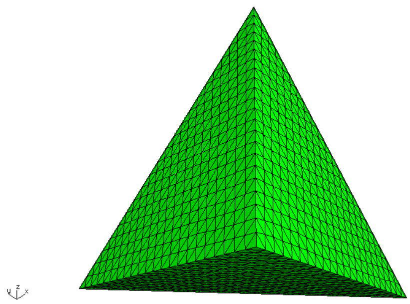


Figure 12: OBJ_Equilateral_Tetrahedron: Discretised object.

The parameter settings to reproduce the example is presented in Table 38.

main.py

Geometry = "tetraEqui.geo"	alpha = 0.01	MeshSize = 2
Order = 3	Start = 2	Finish = 9
Points = 100	Single = False	Omega = 100000
Pod = True	MultiProcessing = True	

Settings.py

CPUs = 8	BigProblem = False	PODPoints = 13
PODTol = 1e-08	OldMesh = False	PlotPod = True
PODErrorBars = False	EddyCurrentTest = False	vtk_output = False
Refine_vtk = False	FolderName = "Default"	Solver = "bddc"
epsi = 1e-12	Maxsteps = 5000	Tolerance = 1e-10
	ngsglobals.msg_level = 3	

Table 38: OBJ_Equilateral_Tetrahedron: data of simulation.

10.2 OBJ_Irregular_Tetrahedron_mu_1

The geometry is a tetrahedron $B_\alpha := \alpha B$. The non-dimensional irregular tetrahedron B is chosen to have vertices at the locations

$$\boldsymbol{v}_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \boldsymbol{v}_2 = \begin{pmatrix} 7 \\ 0 \\ 0 \end{pmatrix}, \quad \boldsymbol{v}_3 = \begin{pmatrix} 5.5 \\ 4.6 \\ 0.0 \end{pmatrix}, \quad \boldsymbol{v}_4 = \begin{pmatrix} 3.3 \\ 2.0 \\ 5.0 \end{pmatrix}. \tag{10.2}$$

The scaling parameter is $\alpha = 0.01\,\mathrm{m}$. The material properties are $\mu_r = 1$ and $\sigma_* = 1 \times 10^7$ S/m. The object B and the region around it to a truncating boundary in the form of the box $[-100, 100]^3$ is discretised by 22 923 unstructured tetrahedral elements. The resulting MPT has six independent coefficients $(\mathcal{M})_{ij}$ at each frequency. The discretised object B is presented in Figure 13 and the simulation is using polynomial order $p = 3$.

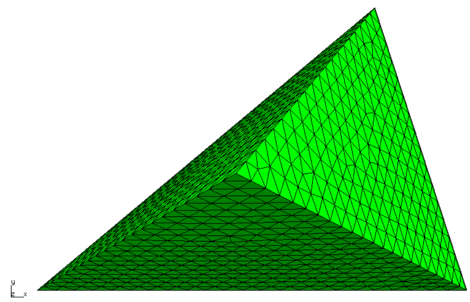


Figure 13: OBJ_Irregular_Tetrahedron_mu_1: Discretised object.

The parameter settings to reproduce the example is presented in Table 38.

`main.py`

`Geometry = "tetraEqui.geo"`	`alpha = 0.01`	`MeshSize = 3`
`Order = 3`	`Start = 2`	`Finish = 9`
`Points = 100`	`Single = False`	`Omega = 100000`
`Pod = True`	`MultiProcessing = True`	

`Settings.py`

`CPUs = 4`	`BigProblem = False`	`PODPoints = 13`
`PODTol = 1e-08`	`OldMesh = False`	`PlotPod = True`
`PODErrorBars = False`	`EddyCurrentTest = False`	`vtk_output = False`
`Refine_vtk = False`	`FolderName = "Default"`	`Solver = "bddc"`
`epsi = 1e-12`	`Maxsteps = 5000`	`Tolerance = 1e-7`
	`ngsglobals.msg_level = 3`	

Table 39: OBJ_Irregular_Tetrahedron_mu_1: data of simulation.

10.3 OBJ_Irregular_Tetrahedron_mu_2_freq_100

The geometry is a tetrahedron $B_\alpha := \alpha B$, the configuration and the mesh are the same as presented in Section 10.2. The material properties are $\mu_r = 2$ and $\sigma_* = 5.96 \times 10^6$ S/m. The parameter settings to reproduce the example is presented in Table 40.

`main.py`

`Geometry = "Tetra.geo"`	`alpha = 0.01`	`MeshSize = 3`
`Order = 3,4`	`Start = 0`	`Finish = 8`
`Points = 100`	`Single = False`	`Omega = 100000`
`Pod = True`	`MultiProcessing = True`	

`Settings.py`

`CPUs = 8`	`BigProblem = False`	`PODPoints = 13`
`PODTol = 1e-08`	`OldMesh = False`	`PlotPod = True`
`PODErrorBars = False`	`EddyCurrentTest = False`	`vtk_output = False`
`Refine_vtk = False`	`FolderName = "Default"`	`Solver = "bddc"`
`epsi = 1e-12`	`Maxsteps = 5000`	`Tolerance = 1e-7`
	`ngsglobals.msg_level = 3`	

Table 40: OBJ_Irregular_Tetrahedron_mu_2_freq_100: data of simulation.

10.4 OBJ_Irregular_Tetrahedron_mu_2_freq_81

The geometry is a tetrahedron $B_\alpha := \alpha B$, the configuration and the mesh are the same as presented in Section 10.3. The object B and the region around it to a truncating boundary in the form of the box $[-100, 100]^3$ is discretised by 21 427 unstructured tetrahedral elements. The parameter settings to reproduce the example is presented in Table 41.

main.py

`Geometry = "Tetra.geo"`	`alpha = 0.01`	`MeshSize = 2`
`Order = 3`	`Start = 2`	`Finish = 8`
`Points = 81`	`Single = False`	`Omega = 100000`
`Pod = True`	`MultiProcessing = True`	

Settings.py

`CPUs = 5`	`BigProblem = False`	`PODPoints = 21`
`PODTol = 1e-04`	`OldMesh = False`	`PlotPod = True`
`PODErrorBars = True`	`EddyCurrentTest = True`	`vtk_output = False`
`Refine_vtk = False`	`FolderName = "Default"`	`Solver = "bddc"`
`epsi = 1e-12`	`Maxsteps = 1500`	`Tolerance = 1e-10`
	`ngsglobals.msg_level = 3`	

Table 41: OBJ_Irregular_Tetrahedron_mu_2_freq_81: data of simulation.

11 Object - Torus

11.1 OBJ_Torus

The geometry is a torus $B_\alpha := \alpha B$. The non-dimensional torus B has major and minor radii, 2 and 1, respectively. The scaling parameter is $\alpha = 0.01\,\mathrm{m}$. The material properties are $\mu_r = 1.5$ and $\sigma_* = 5 \times 10^5$ S/m. The object B and the region around it to a truncating boundary in the form of a sphere with radius 100 is discretised by 32 008 unstructured tetrahedral elements. The resulting MPT has two independent coefficients $(\mathcal{M})_{11} = (\mathcal{M})_{22}$, and $(\mathcal{M})_{33}$ at each frequency. The discretised object B is presented in Figure 14 and the simulation is using polynomial order $p = 3$.

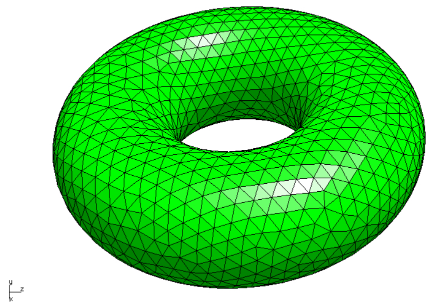


Figure 14: OBJ_Torus: Discretised object.

The parameter settings to reproduce the example is presented in Table 42.

main.py

`Geometry = "Torus.geo"`	`alpha = 0.01`	`MeshSize = 3`
`Order = 3,4`	`Start = 0`	`Finish = 8`
`Points = 100`	`Single = False`	`Omega = 100000`
`Pod = True`	`MultiProcessing = True`	

Settings.py

`CPUs = 16`	`BigProblem = False`	`PODPoints = 20,13`
`PODTol = 1e-08`	`OldMesh = False`	`PlotPod = True`
`PODErrorBars = False`	`EddyCurrentTest = False`	`vtk_output = False`
`Refine_vtk = False`	`FolderName = "Default"`	`Solver = "bddc"`
`epsi = 1e-12`	`Maxsteps = 1500`	`Tolerance = 1e-08`
	`ngsglobals.msg_level = 3`	

Table 42: OBJ_Torus: data of simulation.

12 Object - Two Overlapping Spheres

12.1 OBJ_TwoSpheres

The geometry is a 2-spheres $B_\alpha := \alpha B$. The non-dimensional 2-spheres $B = B_1 \bigcup B_2$ is constructed using two overlapping spheres, such as B_1 is a unitary sphere centred at $(-0.5, 0, 0)$ and B_2 is a unitary sphere centred at $(0.5, 0, 0)$. The scaling parameter is $\alpha = 0.01\,\mathrm{m}$. The material properties are $\mu_r = 1$ and $\sigma_* = 1 \times 10^7$ S/m. The object B and the region around it to a truncating boundary in the form of a sphere with radius 100 is discretised by 47 686 unstructured tetrahedral elements. The resulting MPT has two independent coefficients $(\mathcal{M})_{11} = (\mathcal{M})_{22}$, and $(\mathcal{M})_{33}$ at each frequency. The discretised object B is presented in Figure 15 and the simulation is using polynomial order $p = 3$.

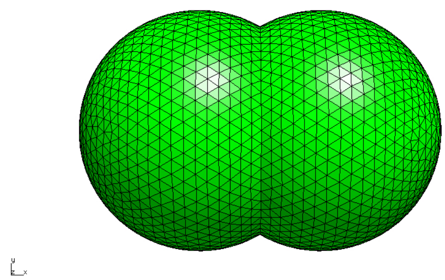


Figure 15: OBJ_TwoSpheres: Discretised object.

The parameter settings to reproduce the example is presented in Table 43.

`main.py`

`Geometry = "twoSpheres.geo"`	`alpha = 0.01`	`MeshSize = 2`
`Order = 3`	`Start = 2`	`Finish = 9`
`Points = 100`	`Single = False`	`Omega = 100000`
`Pod = True`	`MultiProcessing = True`	

`Settings.py`

`CPUs = 8`	`BigProblem = False`	`PODPoints = 13`
`PODTol = 1e-08`	`OldMesh = False`	`PlotPod = True`
`PODErrorBars = False`	`EddyCurrentTest = False`	`vtk_output = False`
`Refine_vtk = False`	`FolderName = "Default"`	`Solver = "bddc"`
`epsi = 1e-12`	`Maxsteps = 1500`	`Tolerance = 1e-10`
	`ngsglobals.msg_level = 3`	

Table 43: OBJ_TwoSpheres: data of simulation.

Acknowledgements

P.D. Ledger and A.A.S. Amad gratefully acknowledges the financial support received from EPSRC in the form of grant EP/R002134/1. B.A. Wilson gratefully acknowledges the financial support received from EPSRC in the form of a DTP studentship with project reference number 2129099.

References

[1] B. A. Wilson, P. D. Ledger, Efficient computation of the magnetic polarizability tensor spectral signature using POD (submitted).

[2] P. D. Ledger, B. A. Wilson, A. A. S. Amad, W. R. B. Lionheart, Identification of metallic objects using spectral mpt signatures: Object characterisation and invariants (in submission).