hpfem-mpt - A tutorial.

P.D. Ledger

Zienkiewicz Centre for Computational Engineering, College of Engineering,

Swansea University, UK

Previous support EPSRC Grant EP/K00428X/1.

Outline

Eddy current model

Eddy current approximation:

$\sqrt{\epsilon_* \mu_*}\alpha\omega \ll 1, \epsilon_*\omega/\sigma_* \ll 1$(rigorous justification involves the topology of the object).

In absence of a conducting object the fields from a coil $\boldsymbol{E}_0, \boldsymbol{H}_0$ satisify

$$\nabla \times \boldsymbol{E}_0 = \mathrm{i}\omega\mu_0\boldsymbol{H}_0, \qquad \nabla \times \boldsymbol{H}_0 = \boldsymbol{J}_0 \qquad \text{in } \mathbb{R}^3$$
$$\nabla \cdot \boldsymbol{E}_0 = 0, \qquad \nabla \cdot \boldsymbol{H}_0 = 0 \qquad \text{in } \mathbb{R}^3$$

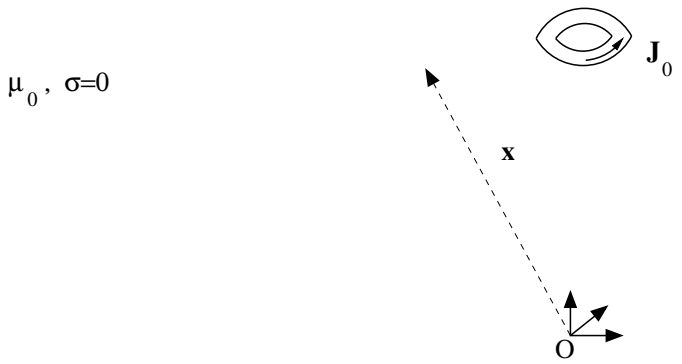
In the presence of a conductor body B_α with

$$\mu_\alpha = \begin{cases} \mu_0 & \text{in } \mathbb{R}^3 \backslash B_\alpha \\ \mu_* & \text{in } B_\alpha \end{cases} \qquad \sigma_\alpha = \begin{cases} 0 & \text{in } \mathbb{R}^3 \backslash B_\alpha \\ \sigma_* & \text{in } B_\alpha \end{cases}$$

the interaction fields $\boldsymbol{E}_\alpha, \boldsymbol{H}_\alpha$ satisfy

$$\nabla \times \boldsymbol{E}_\alpha = \mathrm{i}\omega\mu_\alpha\boldsymbol{H}_\alpha, \qquad \nabla \times \boldsymbol{H}_\alpha = \sigma_\alpha\boldsymbol{E}_\alpha + \boldsymbol{J}_0 \qquad \text{in } \mathbb{R}^3$$
$$\nabla \cdot \boldsymbol{E}_\alpha = 0, \qquad \nabla \cdot \mu_\alpha\boldsymbol{H}_\alpha = 0 \qquad \text{in } \mathbb{R}^3$$

Eddy current model



Eddy current approximation:

$\sqrt{\epsilon_* \mu_*}\alpha\omega \ll 1, \epsilon_*\omega/\sigma_* \ll 1$(rigorous justification involves the topology of the object).

In absence of a conducting object the fields from a coil $\boldsymbol{E}_0, \boldsymbol{H}_0$ satisify

$$\nabla \times \boldsymbol{E}_0 = \mathrm{i}\omega\mu_0 \boldsymbol{H}_0, \qquad \nabla \times \boldsymbol{H}_0 = \boldsymbol{J}_0 \qquad \text{in } \mathbb{R}^3$$
$$\nabla \cdot \boldsymbol{E}_0 = 0, \qquad \nabla \cdot \boldsymbol{H}_0 = 0 \qquad \text{in } \mathbb{R}^3$$

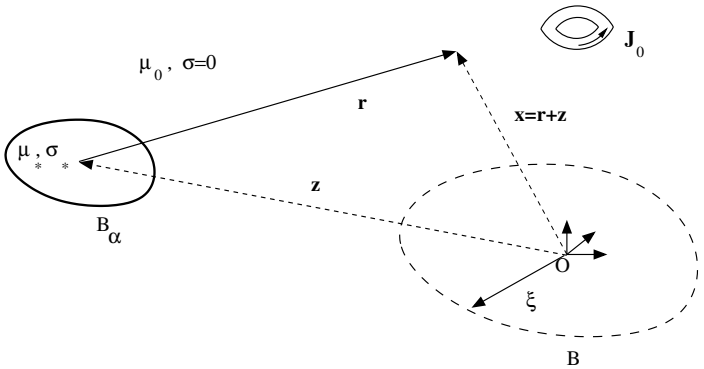
In the presence of a conductor body B_α with

$$\mu_\alpha = \begin{cases} \mu_0 & \text{in } \mathbb{R}^3 \backslash B_\alpha \\ \mu_* & \text{in } B_\alpha \end{cases} \qquad \sigma_\alpha = \begin{cases} 0 & \text{in } \mathbb{R}^3 \backslash B_\alpha \\ \sigma_* & \text{in } B_\alpha \end{cases}$$

the interaction fields $\boldsymbol{E}_\alpha, \boldsymbol{H}_\alpha$ satisfy

$$\nabla \times \boldsymbol{E}_\alpha = \mathrm{i}\omega\mu_\alpha \boldsymbol{H}_\alpha, \qquad \nabla \times \boldsymbol{H}_\alpha = \sigma_\alpha \boldsymbol{E}_\alpha + \boldsymbol{J}_0 \qquad \text{in } \mathbb{R}^3$$
$$\nabla \cdot \boldsymbol{E}_\alpha = 0, \qquad \nabla \cdot \mu_\alpha \boldsymbol{H}_\alpha = 0 \qquad \text{in } \mathbb{R}^3$$

Eddy current model



Eddy current approximation:

$\sqrt{\epsilon_* \mu_*}\alpha\omega \ll 1, \epsilon_*\omega/\sigma_* \ll 1$(rigorous justification involves the topology of the object).

In absence of a conducting object the fields from a coil $\boldsymbol{E}_0, \boldsymbol{H}_0$ satisify

$$\nabla \times \boldsymbol{E}_0 = \mathrm{i}\omega\mu_0\boldsymbol{H}_0, \qquad \nabla \times \boldsymbol{H}_0 = \boldsymbol{J}_0 \qquad \text{in } \mathbb{R}^3$$
$$\nabla \cdot \boldsymbol{E}_0 = 0, \qquad \nabla \cdot \boldsymbol{H}_0 = 0 \qquad \text{in } \mathbb{R}^3$$

In the presence of a conductor body B_α with

$$\mu_\alpha = \begin{cases} \mu_0 & \text{in } \mathbb{R}^3 \backslash B_\alpha \\ \mu_* & \text{in } B_\alpha \end{cases} \qquad \sigma_\alpha = \begin{cases} 0 & \text{in } \mathbb{R}^3 \backslash B_\alpha \\ \sigma_* & \text{in } B_\alpha \end{cases}$$

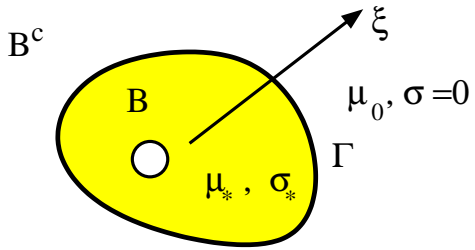
the interaction fields $\boldsymbol{E}_\alpha, \boldsymbol{H}_\alpha$ satisfy

$$\nabla \times \boldsymbol{E}_\alpha = \mathrm{i}\omega\mu_\alpha\boldsymbol{H}_\alpha, \qquad \nabla \times \boldsymbol{H}_\alpha = \sigma_\alpha\boldsymbol{E}_\alpha + \boldsymbol{J}_0 \qquad \text{in } \mathbb{R}^3$$
$$\nabla \cdot \boldsymbol{E}_\alpha = 0, \qquad \nabla \cdot \mu_\alpha\boldsymbol{H}_\alpha = 0 \qquad \text{in } \mathbb{R}^3$$

Summary of key results

By setting $B_\alpha = \alpha B + z$ we have derived that

$$(\boldsymbol{H}_\alpha - \boldsymbol{H}_0)(\boldsymbol{x})_j = \boldsymbol{D}_x^2 G(\boldsymbol{x},\boldsymbol{z})_{jm} \overgtr{\mathcal{M}}_{mi} \boldsymbol{H}_0(\boldsymbol{z})_i + O(\alpha^4),$$



as $\alpha \to 0$ (PDL, WRBL (2015b)) where $G(\boldsymbol{x},\boldsymbol{z}) = 1/(4\pi|\boldsymbol{x}-\boldsymbol{z}|)$.

The rank 2 MPT has coefficients $\overgtr{\mathcal{M}}_{mi} := -\breve{\mathcal{C}}_{mi} + \mathcal{N}_{mi}$ it is **symmetric** and so only 6 are needed where

$$\breve{\mathcal{C}}_{mi} := \frac{\beta}{2}\hat{\boldsymbol{e}}_m \cdot \int_B \boldsymbol{\xi} \times (\boldsymbol{\theta}_i + \hat{\boldsymbol{e}}_i \times \boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} \qquad \mathcal{N}_{mi} := \alpha^3 \left(1 - \frac{\mu_0}{\mu_*}\right) \int_B \left(\hat{\boldsymbol{e}}_m \cdot \hat{\boldsymbol{e}}_i + \frac{1}{2}\hat{\boldsymbol{e}}_m \cdot \nabla_\xi \times \boldsymbol{\theta}_i\right) \mathrm{d}\boldsymbol{\xi}$$

and $\beta = -\frac{\mathrm{i}\nu\alpha^3}{2}$.

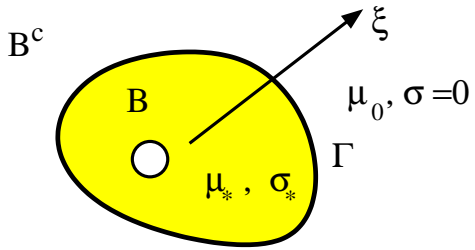
To compute the tensors we solve

$$\begin{aligned}
\nabla_\xi \times \mu_*^{-1} \nabla_\xi \times \boldsymbol{\theta}_i - \mathrm{i}\omega\sigma_* \alpha^2 \boldsymbol{\theta}_i &= \mathrm{i}\omega\sigma_* \alpha^2 \hat{\boldsymbol{e}}_i \times \boldsymbol{\xi} && \text{in } B \\
\nabla_\xi \cdot \boldsymbol{\theta}_i = 0, \qquad \nabla_\xi \times \mu_0^{-1} \nabla_\xi \times \boldsymbol{\theta}_i &= \boldsymbol{0} && \text{in } B^c \\
[\hat{\boldsymbol{n}} \times \boldsymbol{\theta}_i]_\Gamma = \boldsymbol{0}, \qquad [\hat{\boldsymbol{n}} \times \mu^{-1} \nabla_\xi \times \boldsymbol{\theta}_i]_\Gamma &= -2[\mu^{-1}]_\Gamma \hat{\boldsymbol{n}} \times \hat{\boldsymbol{e}}_i && \text{on } \Gamma \\
\boldsymbol{\theta}_i &= O(|\boldsymbol{\xi}|^{-1}) && \text{as } |\boldsymbol{\xi}| \to \infty
\end{aligned}$$

Summary of key results

By setting $B_\alpha = \alpha B + z$ we have derived that

$$(\boldsymbol{H}_\alpha - \boldsymbol{H}_0)(\boldsymbol{x})_j = \boldsymbol{D}^2_x G(\boldsymbol{x},\boldsymbol{z})_{jm} \overgtr{\mathcal{M}}_{mi} \boldsymbol{H}_0(\boldsymbol{z})_i + O(\alpha^4),$$



as $\alpha \to 0$ (PDL, WRBL (2015b)) where $G(\boldsymbol{x},\boldsymbol{z}) = 1/(4\pi|\boldsymbol{x}-\boldsymbol{z}|)$.

The rank 2 MPT has coefficients $\overgtr{\mathcal{M}}_{mi} := -\breve{\mathcal{C}}_{mi} + \mathcal{N}_{mi}$ it is **symmetric** and so only 6 are needed where

$$\breve{\mathcal{C}}_{mi} := \frac{\beta}{2}\hat{\boldsymbol{e}}_m \cdot \int_B \boldsymbol{\xi} \times (\boldsymbol{\theta}_i + \hat{\boldsymbol{e}}_i \times \boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} \qquad \mathcal{N}_{mi} := \alpha^3 \left(1 - \frac{\mu_0}{\mu_*}\right) \int_B \left(\hat{\boldsymbol{e}}_m \cdot \hat{\boldsymbol{e}}_i + \frac{1}{2}\hat{\boldsymbol{e}}_m \cdot \nabla_\xi \times \boldsymbol{\theta}_i\right) \mathrm{d}\boldsymbol{\xi}$$

and $\beta = -\frac{\mathrm{i}\nu\alpha^3}{2}$.

To compute the tensors we solve

$$\begin{aligned}
\nabla_\xi \times \mu_*^{-1} \nabla_\xi \times \boldsymbol{\theta}_i - \mathrm{i}\omega\sigma_* \alpha^2 \boldsymbol{\theta}_i &= \mathrm{i}\omega\sigma_* \alpha^2 \hat{\boldsymbol{e}}_i \times \boldsymbol{\xi} && \text{in } B \\
\nabla_\xi \cdot \boldsymbol{\theta}_i = 0, \qquad \nabla_\xi \times \mu_0^{-1} \nabla_\xi \times \boldsymbol{\theta}_i &= \boldsymbol{0} && \text{in } B^c \\
[\hat{\boldsymbol{n}} \times \boldsymbol{\theta}_i]_\Gamma = \boldsymbol{0}, \qquad [\hat{\boldsymbol{n}} \times \mu^{-1} \nabla_\xi \times \boldsymbol{\theta}_i]_\Gamma &= -2[\mu^{-1}]_\Gamma \hat{\boldsymbol{n}} \times \hat{\boldsymbol{e}}_i && \text{on } \Gamma \\
\boldsymbol{\theta}_i &= O(|\boldsymbol{\xi}|^{-1}) && \text{as } |\boldsymbol{\xi}| \to \infty
\end{aligned}$$

Further details

For further details on MPTs and their properties please see the following:

- PDL, WRBL, “Characterizing the shape and material properties of hidden targets from magnetic induction data”, **IMA Journal of Applied Mathematics**, vol 80, pp 1776-1798, 2015.
- PDL, WRBL “Understanding the magnetic polarizability tensor” , IEEE Trans. Magn. doi: **10.1109/TMAG.2015.2507169** (2016) .
- PDL E-Presentation “Characterising the shape and material properties of hidden conducting targets in metal detection”, **Video**, LMS-EPSRC Symposium on Maxwell’s equations, 2016.
- PDL Presentation “A justification for the rank 2 magnetic polarizability tensor description of conducting objects in metal detection”, **Slides**, CIRCLE, The University of Manchester, 2016.

Outline

Dependencies

`hpfem-mpt` is designed to be an educational and research tool and is fairly self contained.

This version is implemented in `MATLAB`, which is not as fast as my `FORTRAN` version (not available at present), but does offer greater portability as it does not require compilers and the installation of various libraries.

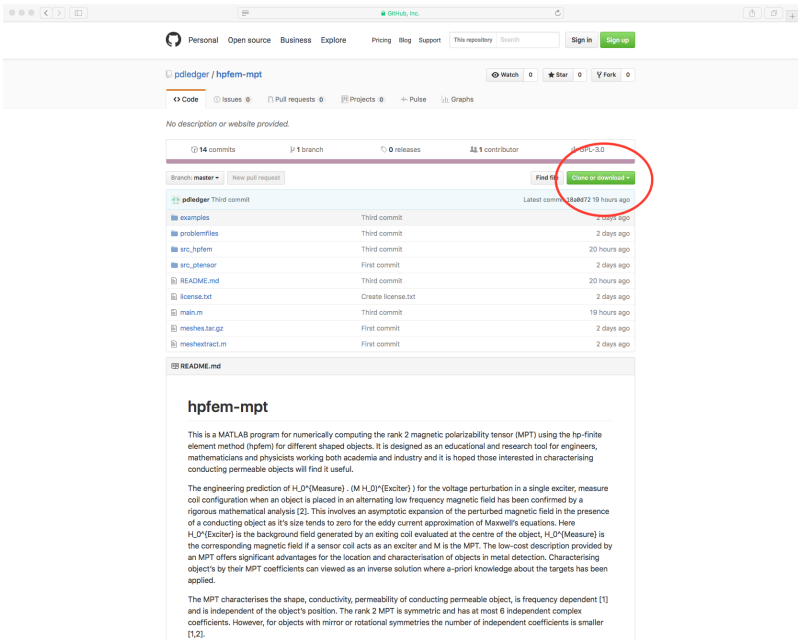
Furthermore, no knowledge of `Makefiles` is required and comes setup with several examples for you to try out.

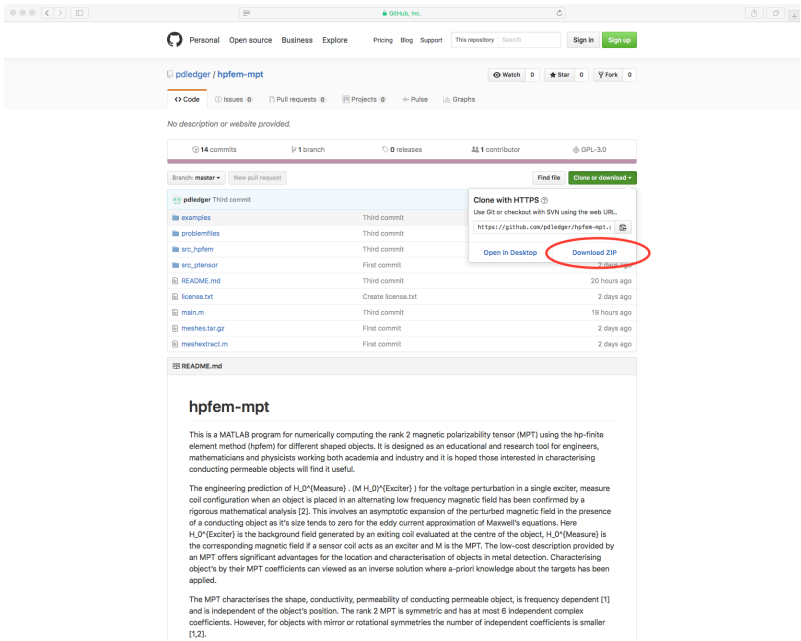
`hpfem-mpt` does however require the following to be installed

- `MATLAB` (tested on version 2015b)
- `NETGEN` Unstructured mesh generator, available **here**.
- `Paraview` Visualisation software, available **here**.

Obtaining the current version

To obtain the current version go to
`https://github.com/pdledger/hpfem-mpt/tree/master` **link**





Click on `Download or Clone` followed by `Download ZIP` to save the repository to your computer and unzip it.

Getting the mesh files ready

A series of example mesh files are included. To prepare them start `MATLAB` and change the working directory to where the unzipped file is stored.

In the `MATLAB` command window type

```
meshextract
```

This will untar and unzip the mesh files and create a folder called `meshes` containing the example meshes.

Outline

Tutorial 1 overview

In this tutorial we will

- Understand how to generate meshes of different mesh densities from a `.geo` file in `NETGEN`;
- Understand how different problems can be selected in `hpfem-mpt`;
- Understand how to change the frequency, size and material parameters of an object;
- Understand how the field and the geometry are discretised differently;
- Understand how to perform run `hpfem-mpt` and perform p-refinement;
- Compare the accuracy of h and p-refinement and the influence on the accuracy of the MPT.

Mesh generation-1

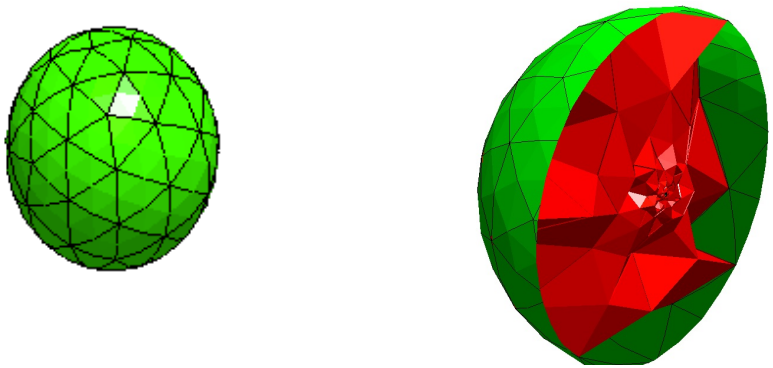
A mesh with a coarse discretisation of 2425 elements is saved in `sphere11.vol` in the `meshes` folder. This mesh is generated in `NETGEN` using the `sphere11.geo` file. It is important to choose the object (here `sphin`) to be of unit size and to note the order of the `tlo`

```
algebraic3d
#
# Example with two sub-domains:  was 10
#
solid sphout = sphere (0, 0, 0; 100) -bco=1;
solid sphin = sphere (0, 0, 0; 1) -bco=2;

solid rest = sphout and not sphin;

tlo rest -transparent -col=[0,0,1];
tlo sphin -col=[1,0,0];
```

as this controls the materials IDs (in this case, 1 is the region outside the sphere and 2 the conducting sphere).

Mesh generation-2

Opening the geomtry file using `File > Load Geometry` using the default settings in `NETGEN` and activating `create mesh` will lead to a mesh with more elements and only a linear geometry approximation.



The mesh of 2425 the meshes folder can be reproduced by using the mesh settings of `coarse` and `second order elements` in the `Meshing Options` palette (`Mesh> Meshing Options`) in `NETGEN`.

Using `second order elements` should always be selected if the geometry is curved as this improves the accuracy of the geometry.

Coarse meshes are generally sufficient since `hpfem-mpt` includes the possibility to use high order elements built on this coarse tetrahedral mesh.

Mesh generation-3

Boundaries are automatically detected (provided the object and far field are not connected) and the default settings do not need to changed.

To save a mesh select `File > Save Mesh` and save it as a (different) `.vol` file in the `meshes` directory.

Should you wish to generate finer meshes, this can easily controlled via the `Mesh granularity` drop down menu in the `Meshing Options` palette in `NETGEN`

`NEGEN` does offer surface and volume mesh checking (see `Mesh> Check Surface Mesh` and `Mesh> Check Volume Mesh`), but these only work provided that `Second order elements` are not selected (in Version 4.9.13). So if generating a mesh for a new geometry, first do it with linear geometry, check the mesh, and then repeat the mesh generation for `Second order elements`, but do not rerun the checks.

Selecting the problem

`hpfem-mpt` offers a set of 10 different examples

In `main.m` they are listed as

```
% problem 1 = conducting magnetic sphere
% problem 2 = conducting magnetic spheroid
% problem 3 = nested sphere conducting and magnetic
% problem 4 = conducting magnetic Remington shell casing
% problem 5 = simple steel gun L shape
% problem 6 = 72 anologue detonator
% problem 7 = belt buckle
% problem 8 = coin
% problem 9 = conducting magnetic torus
% problem 10 = Ansys Mesh (Sphere)
```

and the problem is selected using the flag `problem =1` (default), which controls the problem file to be used. In this case `problem1.m` in the `problemfiles` directory.

Controlling the polynomial order of the discretisation

The other important flag in `main.m` is `pm`, which controls the polynomial order of the elements.

The default is `pm=0` which means the lowest order Nédélec elements with constant tangential component approximation on the edges of the elements will be used.

Changing this to `pm=1`, means Nédélec elements with a linear tangential component approximation will be used, `pm=2`, means Nédélec elements with a quadratic tangential component approximation will be used, ...

In principal the code allows `pm` to be repeatability increased. But for practical reasons, going beyond `p=5` is not recommended as it will lead to excessive long run times.

It is important to note the field approximation, controlled by `pm` *is different to the geometry approximation. This is because in many cases the geometry is flat sided, but yet we still want to use a high approximation to the transmission problem.*

The problem file-1

The problem file `problem1.m` contained in the `problemfiles` directory has already been set up with default output options, material parameters, boundary conditions and source terms and other flags.

Boundary conditions and source terms and other standard flags are handled in

```
...
% include standard options (include BC, src definition etc)
probdata=[];
probdata=standoptions(probdata);
...
```

and shouldn't ever need to be changed.

The part of `problem1.m` that has

```
...
job data-----------------------------------------------------------
job = 'sphere11'    % Job Name
meshtype = 3;       % Mesh type 1= FLITE , 2=NG (old style), 3=NG (new style)
..
```

controls with `job` which mesh will be used (minus the `.vol` extension). You can select the sphere you've just generated if you wish.

The problem file-2

```
...
%material data-----------------------------------------------------------------
muz = 1.256637061435917e-06; % Mu_z
epz = 0;                     % Ep_z
omega =133.5;            % Omega
...
```

specifies μ_0 (in H/m) and ω (in rad/s). Changes to the frequency are possible here.

For each subdomain we need to specify the relative permeability μ_r and conductivity σ. For non-conducting regions a conductivity of O(0.1) is specified since the regularisation term is computed in the form $\varepsilon = \sigma\mu_0\omega$ for this subdomain and hence the regularisation parameter $\varepsilon \sim O(10^{-5})$.

For the subdomain(s) corresponding to the conducting object we can change μ_r and σ (in S/m) ....

The problem file-3

```
...
% For each subdomain specify parameters
% Mu_r, Ep_r, Sigma, J
% Mat 1
mu(1) = 1;
epl(1) = 0;
sigma(1) = 0.1;
jsrc(1,1:3) = [0  0  0];

% Mat 2
mu(2) = 1.5
epl(2) = 0;
sigma(2) = 5.96e7;
jsrc(2,1:3) = [0  0  0];

% specify the material to be used a conductor conductors
% ie specify regions where gradients basis functions to be included
matcond=[2];
...
```

specifies the materials for each subdomain (according to the order of the TLOs in the geometry file). The array `matcond` specifies which subdomain numbers are to be treated as conductors (sometimes an object may be made up of several TLOs).

Problem file-4

Recall that our spherical conducting object was assumed to have unit radius in the mesh file.

```
...
% In this case the mesh is for a unit sized object, it must be scaled (and
% repositioned)
delta = 0.01;  % Object size
shift=[0 0 0]; % Object shift
...
```

control's the object's size. `delta=0.01` means that each coordinate direction will be scaled by this factor.

In other words we can compute the MPT for a sphere of radius 0.01m, but computations will be done on the mesh for the unit radius sphere.

This setup means we only need to generate the mesh once to compute MPTs for spheres with a range of radii.

Problem file-5

The default geometry approximation (which will use quadratic geometry if available in the mesh file, linear otherwise) is set up in `standoptions` call we saw earlier.

In the case of a sphere we want to use a high order geometry approximation, due to its simplicity (and in order confirm exponential convergence of the MPT using p-refinement).

```
...
% Blending Function Info (overwrite defaults)
gorder =4;          % order (if gorder > 0, quadlin =2 required)
g1 = 2;             % use exact geometry for a sphere
rin(1) = 1;         % raduis
sufv(1)= 5;         % surfaceval
gag = 2;            % Goagain  % 1- go again 2- once
svchk = 1;          % Check surface/volume as expecting a sphere
...
```

The above tells the program to use a 5th order polynomial approximation for the geometry spherical object (Recall the field approximation is separate).

Running hpfem-mpt

Let us assume we have left the options as their default values, `pm=0`, `problem=1` etc.

To run the program, select the working directory to be the root of the program and type

```
main
```

in the command window. The program will read then

- Read the mesh, identify boundaries edges and faces and number the unknowns;
- Compute the geometry approximation;
- Compute boundary condition values, assemble the matrices;
- Solve the linear system;
- Compute the MPT;
- Write output files.

Interpreting the output -1

```
...
The number of edge based unknowns is 2767
The total number of edge and face unknowns is2767
The total number of PEC edge and face knowns is180
The total number of edge face and interior unknowns is2767
...
```

Tell us about the size of the computation.

```
...
The relative error in the surface of the sphere is3.7288e-06
The relative error in the volume of the sphere is5.6503e-06
...
```

Tell us about the accuracy of the geometry approximation.

```
...
solving linear system
completed LU decomposition
System solved successfully
Number of iterations3
...
```

Tell us about the linear equation solve.

Interpreting the output -2

```
ptensor =
   1.6751e-06 + 6.4681e-07i   1.7022e-09 + 1.2589e-09i  -7.4196e-10 - 8.3201e-10i
   1.7022e-09 + 1.2589e-09i   1.6787e-06 + 6.4952e-07i  -2.3863e-11 - 9.6766e-11i
  -7.4196e-10 - 8.3201e-10i  -2.3863e-11 - 9.6766e-11i   1.6782e-06 + 6.4879e-07i

The exact polarisation tensor is diagonal with coefficient 1.7078e-06+6.8041e-07i
The relative error in the computed tensor=0.026095
```

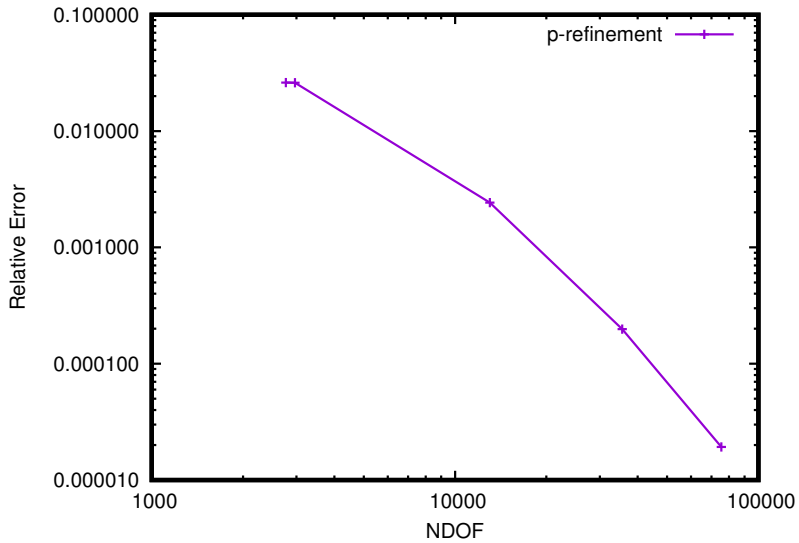
Tell us about the computed tensor and, in this case, the expected analytical result.

Let us suppose we were not happy with this approximation and wanted to improve it. We can do so by increasing `pm`

Recording the total number of unknowns the accuracy of the computed tensor we can plot a graph.

Performing p-refinement

Performing calculations with `pm=0`, `pm=1`, `pm=2`, `pm=3`, `pm=4` with a fixed mesh in sequence leads to p-refinement and displaying the results as a graph:

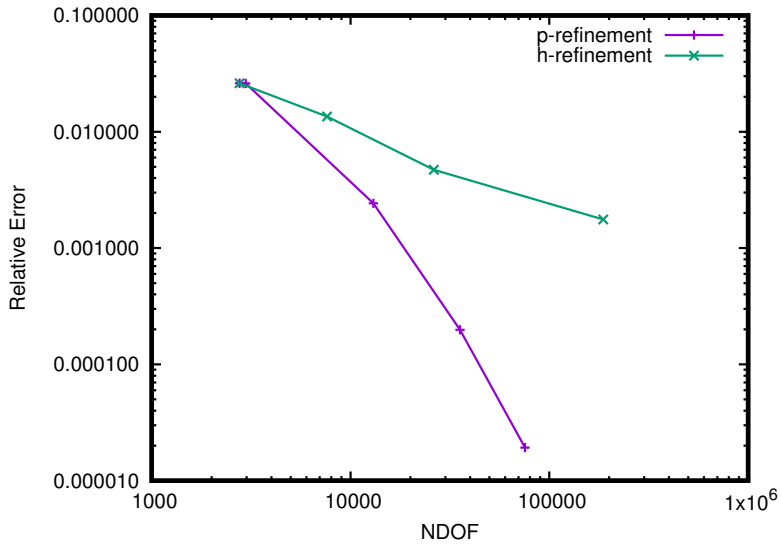


We see that performing p-refinement leads to a downward sloping curve and exponential convergence of the tensor coefficients.

Note that going from `pm=0`, `pm=1` often does not change the accuracy, this is because only the approximation of the field rather its curl is improved for this polynomial increment.

Performing h-refinement

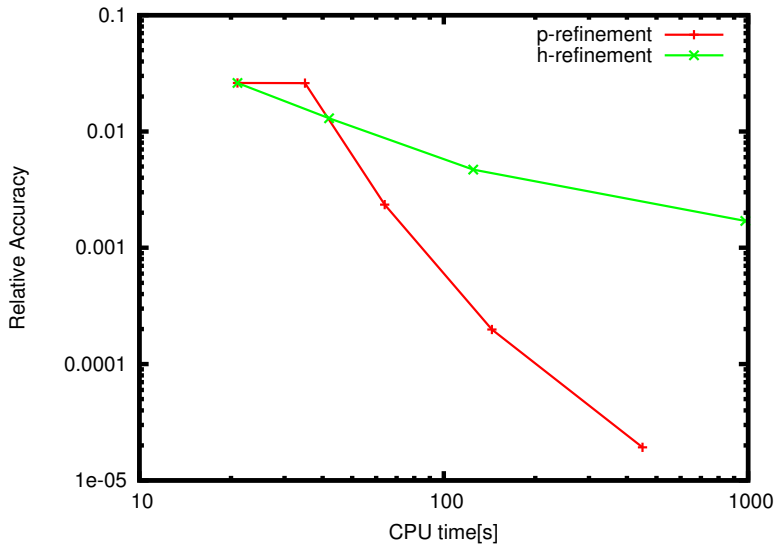
As a comparison, we can return to `NETGEN` and generate and saves meshes with granularities of `Coarse` (current), `Moderate`, `Fine` and `Very Fine` (provided your machine has enough RAM) and rerun `hpfem-mpt` with `pm=0` in each case. Plotting out the results on the same graph gives: h-refinement"



We see that performing h-refinement leads to algebraic convergence and that the previous p-refinement outperforms this.

Comparing CPU time

By recording the CPU time for both p– and h–refinements it is also possible to compare CPU time:



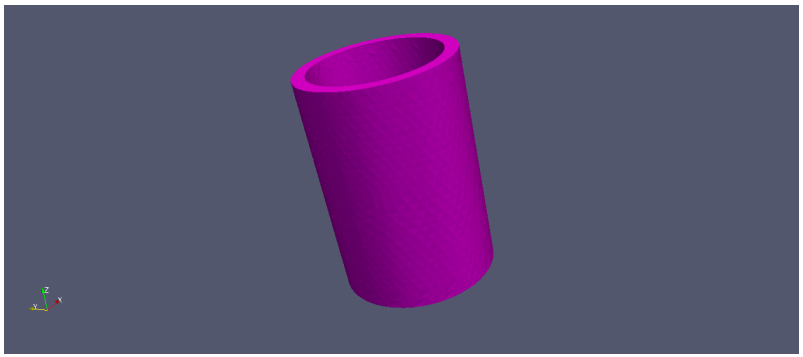
Again p– outperforms h–refinement. Further savings may also be possible.

Outline

Tutorial 2 overview

In this tutorial we will

- Understand how to work with objects made up of multiple subdomains;
- Understand how to check convergence for problems without analytical solution;
- Visualisation of solutions to the transmission problem using `paraview`;



We will do this in the context of an idealised landmine component.

Mesh generation

A mesh of 15 856 elements is saved in `72analogue2.vol` in the `meshes` folder. This mesh is generated in `NETGEN` using the `72analogue.geo` file. It is important to note the ordering of the `tlo`

```
algebraic3d

solid boxout = orthobrick (-200, -200, -200; 200, 200, 200) -bco=2;
solid analogue1 = cylinder ( 0, 0, -0.375; 0, 0, 0.475; 0.3 )
        and plane (0, 0, -0.375; 0, 0, -1)
        and plane (0, 0, 0.475; 0, 0, 1) -bco=1;
solid analogue2= cone ( 0, 0, -0.375; 0.2; 0, 0, 0.475; 0.25 )
        and plane (0, 0, -0.375; 0, 0, -1)
        and plane (0, 0, 0.475; 0, 0, 1) -bco=1;

solid rest = boxout and not analogue1;
solid analogue=analogue1 and not analogue2;

tlo rest -transparent -col=[0,0,1];
tlo analogue2 -transparent -col=[0,0,1];
tlo analogue -col=[1,0,0];
```

In this case there are 3 subdomains, the first and second are free space regions. The third region is the object, which is of unit (O(1)) size.

The mesh can be reproduced in `NETGEN` with the options of `moderate` mesh granularity and `Second order elements`.

Problem file

In `problem1landmine.m` we highlight how the materials are described

```
% Mu_r, Ep_r, Sigma, J
% Free space
mu(1) = 1;
epl(1) = 0;
sigma(1) = 0.1;
jsrc(1,1:3) = [0  0  0];
mu(2) = 1;
epl(2) = 0;
sigma(2) = 0.1;
jsrc(2,1:3) = [0  0  0];

% Object
mu(3) = 1.000022;
epl(3) = 0;
sigma(3) = 3.5e7;
jsrc(3,1:3) = [0  0  0];

% specify the material to be used a conductor conductors
% ie specify regions where gradients basis functions to be included
matcond=[3];
```

For the first two subdomains material parameters are assigned based on them being free space (as per the geometry description). The third subdomain is the object and the parameters are set accordingly. The object size is set to be

```
...
delta = 0.01;  % Object size
...
```

Obtaining a converged solution-1

As in the previous tutorial we use p refinement to reach a converged solution. Selecting `problem=6` in `main.m` and running with `pm=0` we get

```
ptensor =
  -6.4540e-13 + 9.4842e-10i   2.0467e-15 - 5.1568e-13i  -1.3046e-15 - 7.6587e-16i
   2.0467e-15 - 5.1568e-13i  -6.5501e-13 + 9.5183e-10i  -2.6143e-15 + 4.3374e-16i
  -1.3046e-15 - 7.6587e-16i  -2.6143e-15 + 4.3374e-16i  -1.7662e-12 + 1.0728e-09i
```

With `pm=1`

```
 ptensor =
  -5.3017e-13 + 9.2072e-10i  -1.9624e-15 + 3.5270e-15i  -2.9812e-15 - 1.0256e-15i
  -1.9626e-15 + 3.5270e-15i  -5.2463e-13 + 9.2070e-10i   2.6514e-15 + 2.4742e-16i
  -2.9812e-15 - 1.0256e-15i   2.6514e-15 + 2.4742e-16i  -1.7662e-12 + 1.0728e-09i
```

With `pm=2`

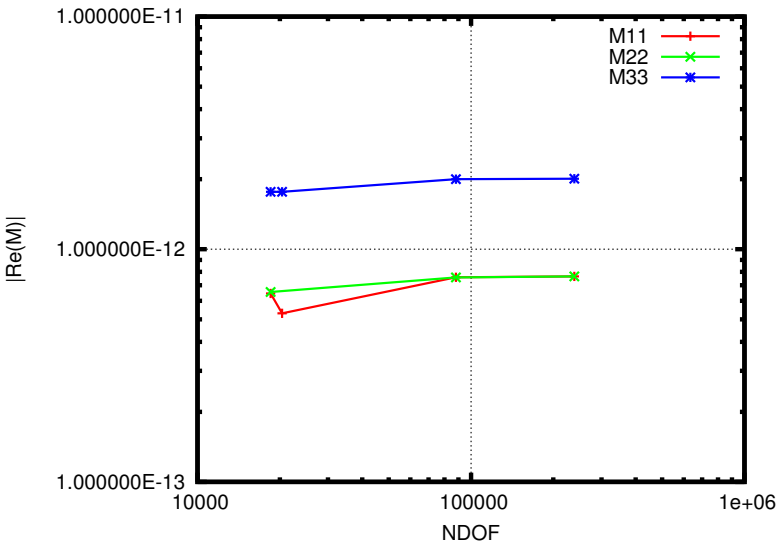
```
ptensor =
  -7.5839e-13 + 9.2023e-10i  -1.2124e-16 + 2.4722e-16i  -1.2212e-17 - 1.0327e-15i
  -1.2111e-16 + 2.4722e-16i  -7.5796e-13 + 9.2023e-10i  -1.5706e-16 + 2.0011e-16i
  -1.2287e-17 - 1.0327e-15i  -1.5677e-16 + 1.9998e-16i  -2.0075e-12 + 1.0728e-09i
```

Obtaining a converged solution-2

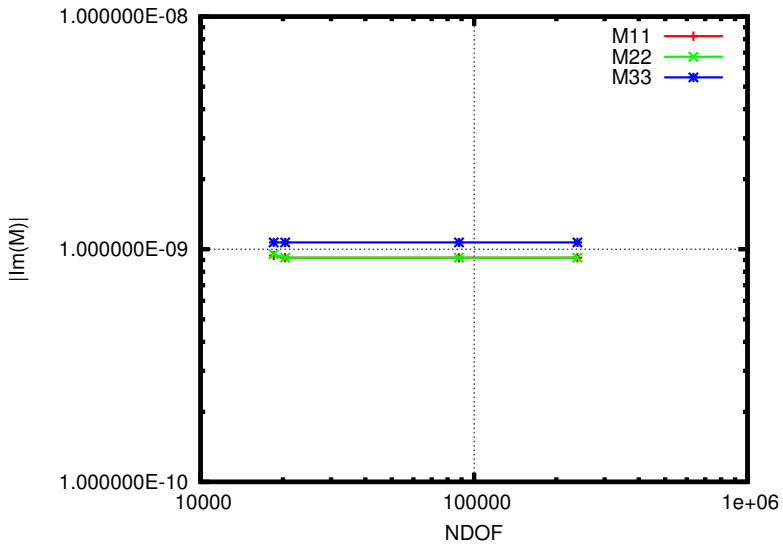
With `pm=3`

```
ptensor =
 -7.6483e-13 + 9.2021e-10i  -1.6239e-17 + 4.2696e-16i   1.0980e-17 - 1.0319e-15i
  -1.6228e-17 + 4.2694e-16i  -7.6478e-13 + 9.2021e-10i  -1.6600e-17 + 2.0691e-16i
   1.0995e-17 - 1.0319e-15i  -1.6620e-17 + 2.0691e-16i  -2.0118e-12 + 1.0728e-09i
```

In this case there is no analytical solution with which to make comparisons against. Instead we look for convergence of the MPT coefficients,.



$|\mathrm{Re}(\overleftrightarrow{\mathcal{M}})|$



$|\mathrm{Im}(\overleftrightarrow{\mathcal{M}})|$

Obtaining a converged solution-3

Although we could go higher with p (at greater computational expense) we see that there are only small changes in $|\mathrm{Re}(\overleftrightarrow{\mathcal{M}})|$ coefficients between $p = 2$ and $p = 3$ (for $|\mathrm{Im}(\overleftrightarrow{\mathcal{M}})|$ are already converged at $p = 1$). We deduce the solution has converged at $p = 2$.

As in the case of the sphere, the tensor is approximately diagonal with the off-diagonal elements several orders of magnitude smaller than the diagonal ones. Rotational and reflectional symmetries tell us that the MPT for this object is described by two independent complex coefficients and this is what we see in this example.

The two independent coefficients are

$$\overleftrightarrow{\mathcal{M}}_{11} = \overleftrightarrow{\mathcal{M}}_{22} = -7.64 \times 10^{-13} + 9.20 \times 10^{-10} i \qquad \overleftrightarrow{\mathcal{M}}_{33} = -2.01 \times 10^{-12} + 1.07 \times 10^{-9} i$$

Visualisation in Paraview-1

Provided that the flag is appropriately set in the problem file

```
...
% output the VTK file
% vtkoption = 0 - do not output
% vtkoption = 1 - output
 vtkoption =1;
```

Provided this flag is set the program will output a `[job].vtm` and multiple `[job]_#.vtu` files with `job` being specified in the problem file. A separate `.vtu` file is created for each subdomain in the mesh and all are referenced by the `[job].vtu` file.

These files contain the solution evaluated on a subdivided mesh for the purpose of visualisation.

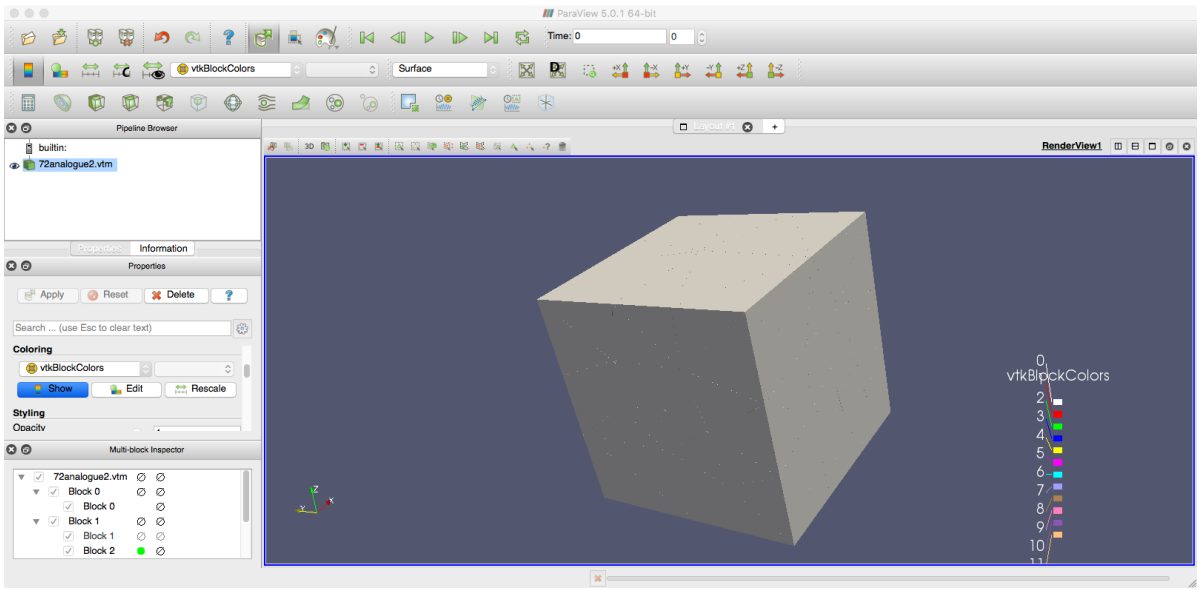
It is necessary to first subdivide the mesh as the visualisation provided by `Paraview` assumes a low order discretisation and to visualise the high-order solutions produced by `hpfem-mpt` correctly they must first be evaluated on a subdivided mesh.

The level of sub-division is controlled by `pm` (upto some prescribed maximum) and uses the curved geometry information received from the mesh.

Visualisation in Paraview-2

In `Paraview` select `File->Open` and navigate to the root directory of `hpfem-mpt` where you should see the `[job].vtu` file listed.

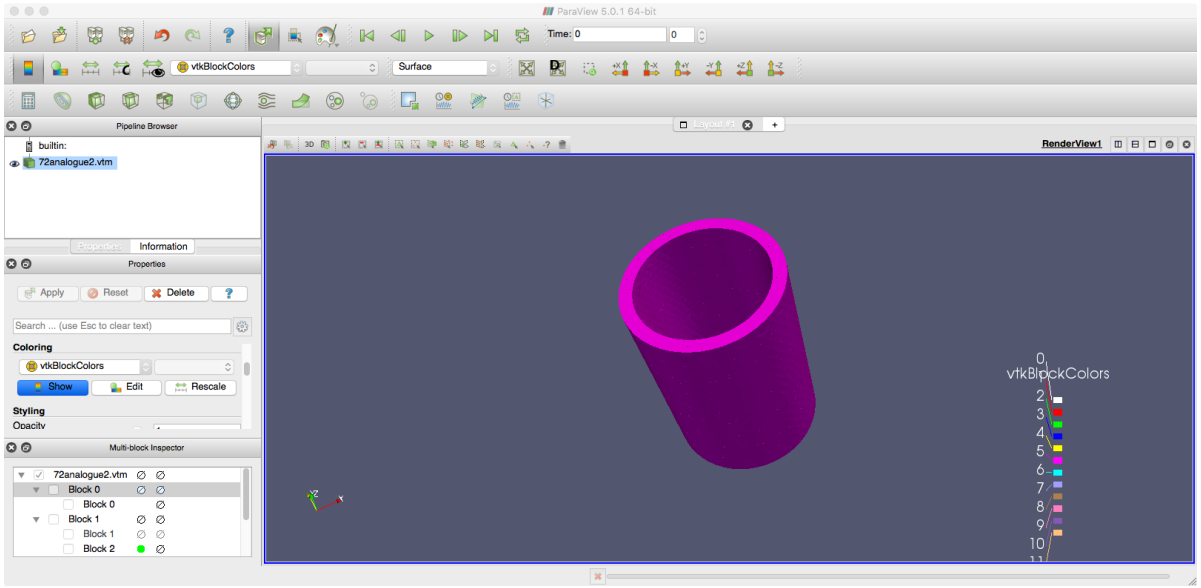
It is important to make sure `Information`, `Multi-block inspector`, `Pipeline Browser` and `Properties` are all available. To do this go to `View` and select the appropriate options. Then click `Apply`.



We are probably most interested in viewing the fields on/in the object so we need to select and deselect appropriate blocks. In `Paraview` the block numbers are numbered one less than the subdomain numbers in `Netgen` and `hpfem-mpt`

Visualisation in Paraview-3

Deselecting the blocks `0` and `1` and zooming on the object we see

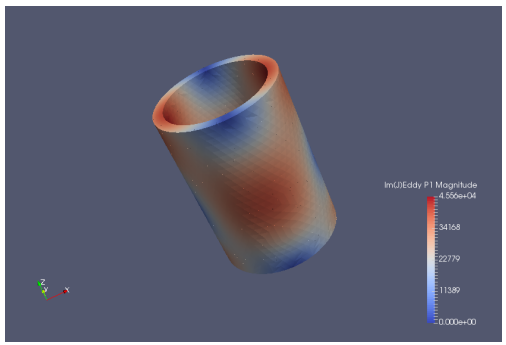


By choosing different options under `coloring` it is possible to visualise a selection of fields from each of the 3 different transmission problems.

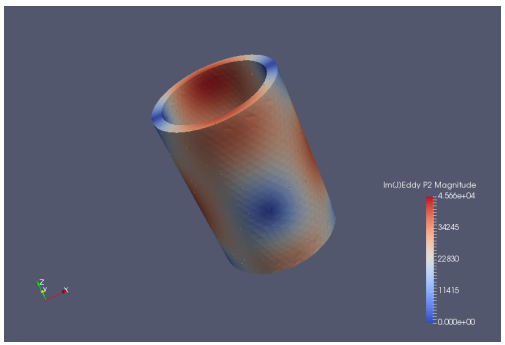
Real and Imaginary parts of the magnetic type field $\nabla \times \boldsymbol{\theta}_i$ (and eddy type current $\sigma\boldsymbol{\theta}_i$) for each of the three problems $i = 1, 2, 3$ are available.

Visualisation in Paraview-3

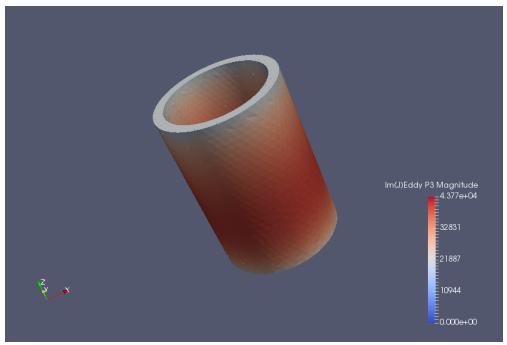
As an example we show $|\mathrm{Im}(\sigma\boldsymbol{\theta}_i)|$ for $i = 1, 2, 3$ on the surface of the object, below



$|\mathrm{Im}(\sigma\boldsymbol{\theta}_1)$



$|\mathrm{Im}(\sigma\boldsymbol{\theta}_2)|$



$|\mathrm{Im}(\sigma\boldsymbol{\theta}_3)|$

Many more visualisation options are available (including vector quiver plots, cutting planes etc). Please see the `Paraview` documentation to find out more.

Other examples

Recall that in `main.m` 10 examples are available

```
% problem 1 = conducting magnetic sphere
% problem 2 = conducting magnetic spheroid
% problem 3 = nested sphere conducting and magnetic
% problem 4 = conducting magnetic Remington shell casing
% problem 5 = simple steel gun L shape
% problem 6 = 72 anologue detonator
% problem 7 = belt buckle
% problem 8 = coin
% problem 9 = conducting magnetic torus
% problem 10 = Ansys Mesh (Sphere)
```

and the results can compared with the values stored in the `examples` folder (note there may be small differences in values due to recent changes in `hpfem-mpt`).

The files `p.0.dat`, `p.1.dat` etc in the folders for the different objects refer to the MPT coefficients for different order elements $p = 0$, $p = 1$ etc.

Outline

Tutorial 3 Overview

In this tutorial we will

- Reiterate the key steps to create meshes with `NETGEN` for new geometries.
- Using `problemfile_template.m` to create problem files for new geometries

Mesh generation recap-1

NETGEN includes a range of topological primitives that can be combined to create a range of interesting shaped objects.

- Blocks;
- Spheres;
- Cones;
- Ellipsoids;
- Torii;
- Planes ....

As we have already seen these can be multiply connected and made up of several different subdomains.

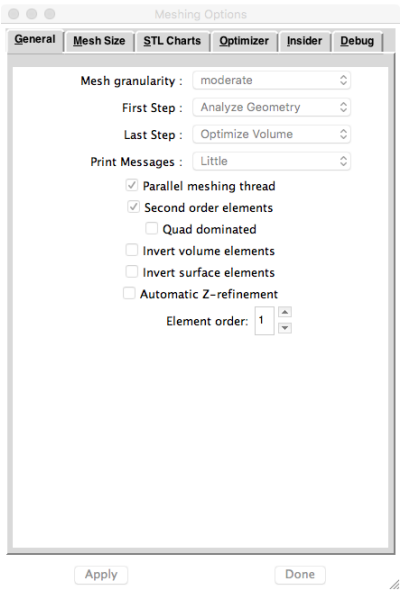
When creating the geometry for the object, it should have outer dimensions that are of $O(1)$ and be placed in a region which is truncated by an outer boundary which is a sphere/box/or other convex shape. Typically this should be chosen to have dimensions $O(100)$ to minimise the effect of approximate far field boundary conditions.

The geometry description is first saved in a `.geo` file that can is then read by NETGEN. For details see the NETGEN documentation.

Mesh generation recap-2

Once loaded in `NETGEN` it is best to first mesh the geometry using the default first order elements and the run the surface and volume mesh checking (see `Mesh> Check Surface Mesh` and `Mesh> Check Volume Mesh`) as discussed in tutorial 1.

Provided these checks are successful, the mesh can be regenerated using `Second order elements` and the desired level of refinement using `Mesh> Meshing Options`.



The mesh should then be saved as a `.vol` file in the `meshes` directory.

Problem file template - filling in the gaps -1

A problem file template `problemfile_template.m` is available in the `problemfiles` directory for you to edit to include the options for your own geometry.

The default options will use quadratic geometry (if available in the mesh file, linear otherwise) and apply the block-Jacobi preconditioner for solving the linear equation system.

The parts requiring input are highlighted in the file. The first is

```
%job data-----------------------------------------------------------------
% Input needed here
job = '%insert job name here%'    % Job Name
% eg  job='sphere11' if the mesh file is sphere11.vol
meshtype = 3;                     % Mesh type 1= FLITE , 2=NG (old style), 3=NG (new style)
% no change to mesh type required if using NETGEN.
```

where the job name (determined by the `.vol` file) should be specified. The next is

```
%Input needed here,
omega = %insert frequency in rad/s;
% eg if omega is 133.5 rad/s enter
% omega =133.5;
```

where ω in (rad/s) should be specified.

Problem file template - filling in the gaps-2

Next, the material parameters for each subdomain (according to the TLO list in your `.geo` file) should be specified

```
% Input needed here
% For each subdomain specify parameters
% Mu_r, Ep_r, Sigma, J
% Mat 1

%eg if the first subdomain is non-conducting enter
%mu(1) = 1;
%epl(1) = 0;
%sigma(1) = 0.1;
%jsrc(1,1:3) = [0  0  0];

% and the second subdomain is the conducting object with mu_r =1.5 and
% sigma =5.96e7 S/m enter
%mu(2) = 1.5
%epl(2) = 0;
%sigma(2) = 5.96e7;
%jsrc(2,1:3) = [0  0  0];


% Input needed here
% specify the material to be used a conductor conductors
% ie specify regions where gradients basis functions to be included
matcond= % insert an array contain the list of conductings subdomains.
% eg if the only conducting subdomain is the second one enter
% matcond=[2];
```

and the subdomains which make up the object listed in `matcond`,

Problem file template - filling in the gaps-3

Finally, the object size scaling parameter should be entered

```
% Input needed here
% In this case the mesh is for a unit sized object, it must be scaled (and
% repositioned)
delta = % enter the object scaling
% specify what scaling to apply to go from the O(1) object specified in the
% mesh to the actual object. Eg if a conducting object of O(1) is specified
% in the mesh but the true object has dimension of O(0.01) enter
%delta = 0.01;  % Object size
```

The final stage is to save your modified `yourproblemfile.m` and include this as an option in `main.m` and to perform a p–convergence study.

Happy computing!