

magnetic resonance integral equation



# User's Guide

## $\beta$ version

Beta Version, Spring 2015

**MARIE - magnetic resonance integral equation suite.  
A prototype MATLAB open source software for  
the fast electromagnetic analysis of MRI systems**

MARIE - Copyright (C) 2015, Jorge Fernández Villena / Athanasios G. Polimeridis  
RLE Computational Prototyping Group, Massachusetts Institute of Technology  
contact: [jvillena@mit.edu](mailto:jvillena@mit.edu) / [thanos\\_p@mit.edu](mailto:thanos_p@mit.edu)

- ◆ **This document introduces**
  - ◆ how to install and prepare the code to run
  - ◆ how to call the GUI
  - ◆ the main functionalities of the GUI
    - ◆ overview of the data structures
    - ◆ each functionality will be illustrated and explained with screenshots

- ◆ **Installation**
- ◆ **MARIE's GUI overview**
  - ◆ File formats and data structures
  - ◆ Coil generator, load and related options
  - ◆ Body generator, load and related options
  - ◆ Simulation and Analysis
  - ◆ Visualization

# installation

## ◆ Requirements

- ◆ MATLAB
- ◆ MARIE is ready to use
  - ◆ extract MARIE zip files into your desired folder
  - ◆ open MATLAB and change into MARIE folder
  - ◆ run **marie.m** script to call the GUI



A screenshot of a MATLAB Command Window. The title bar says "Command Window". In the window, the text "fx >> marie" is visible, with the cursor positioned at the end of the command "marie".

## ◆ MARIE package includes some mex functions

- ◆ Not a requirement, but improve efficiency
- ◆ mex functions can be compiled: call **MARIE\_install.m**
  - ◆ requires C/C++ compiler, preferably with OPENMP compatibility
  - ◆ note: to set up the compiler in MATLAB call **mex - setup**
  - ◆ note: openmp flag for compilation can be added to mexopts file

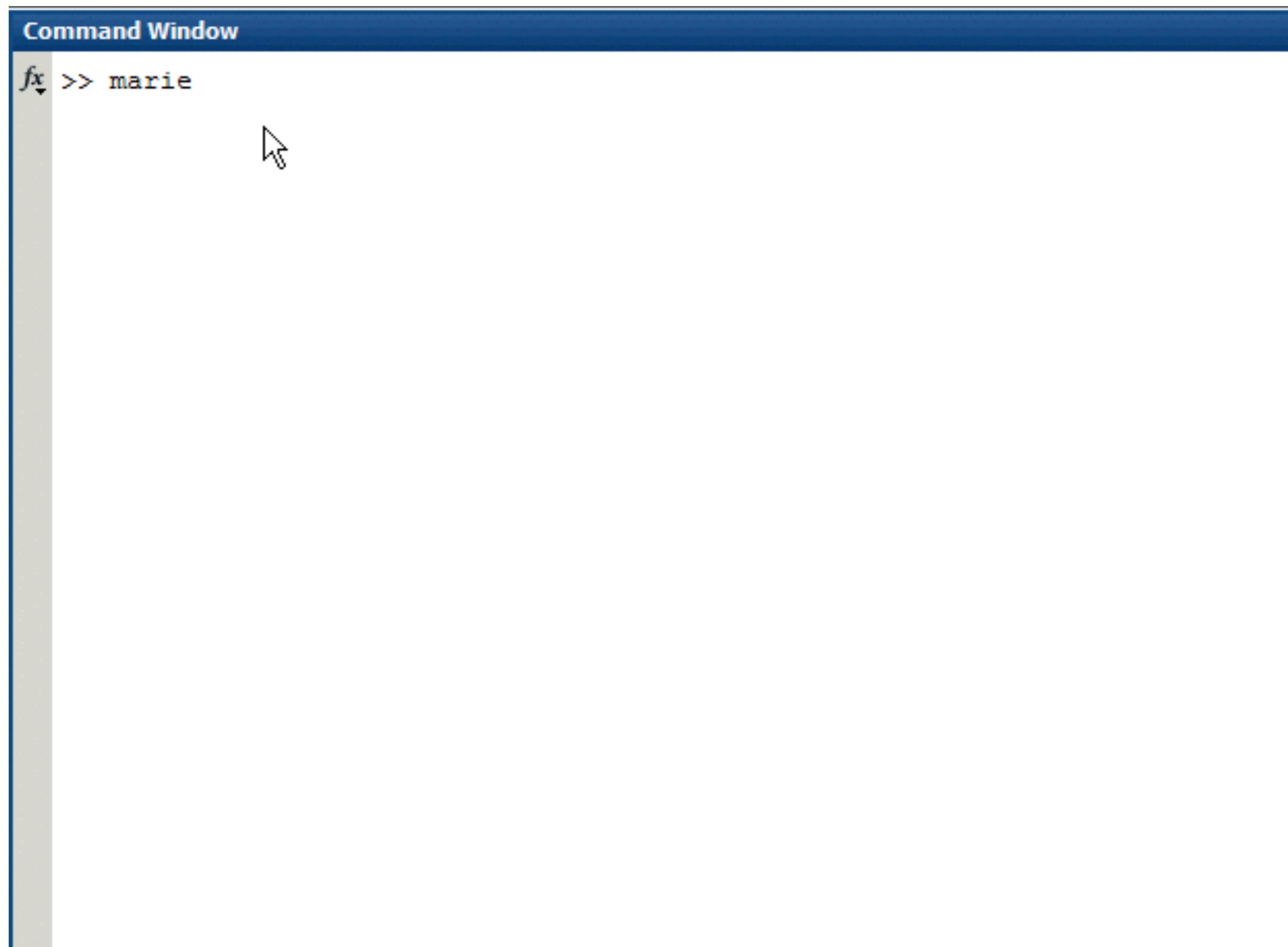
```
Command Window

>> mex -setup
MEX configured to use 'Microsoft Windows SDK 7.1 (C)' for C language compilation.
Warning: The MATLAB C and Fortran API has changed to support MATLAB
variables with more than 2^32-1 elements. In the near future
you will be required to update your code to utilize the
new API. You can find more information about this at:
http://www.mathworks.com/help/matlab/matlab\_external/upgrading-mex-files-to-use-64-bit-api.html.

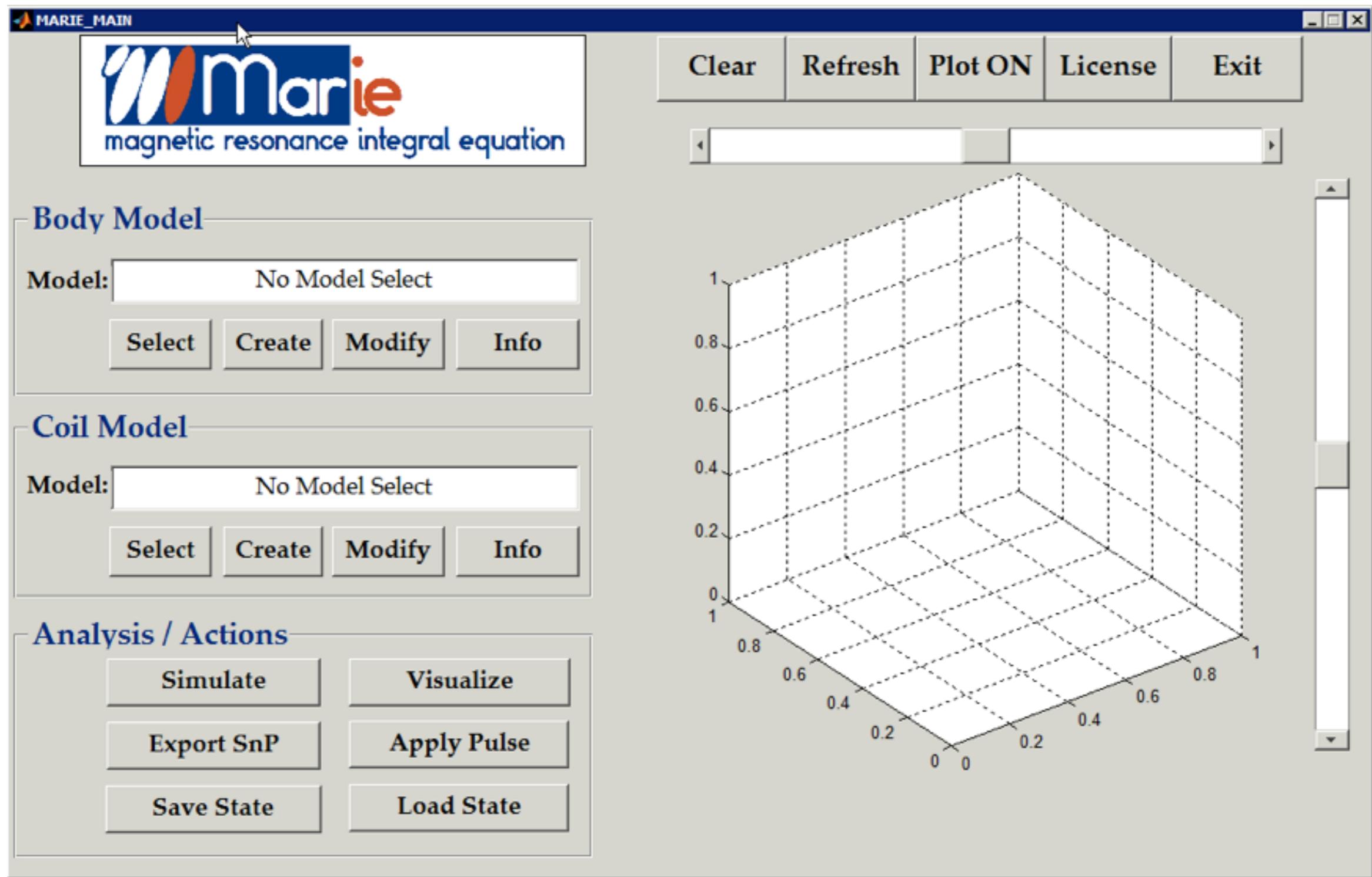
To choose a different language, select one from the following:
mex -setup C++
mex -setup FORTRAN
>>
>>
>>
fx >> MARIE_install
```

graphic user interface

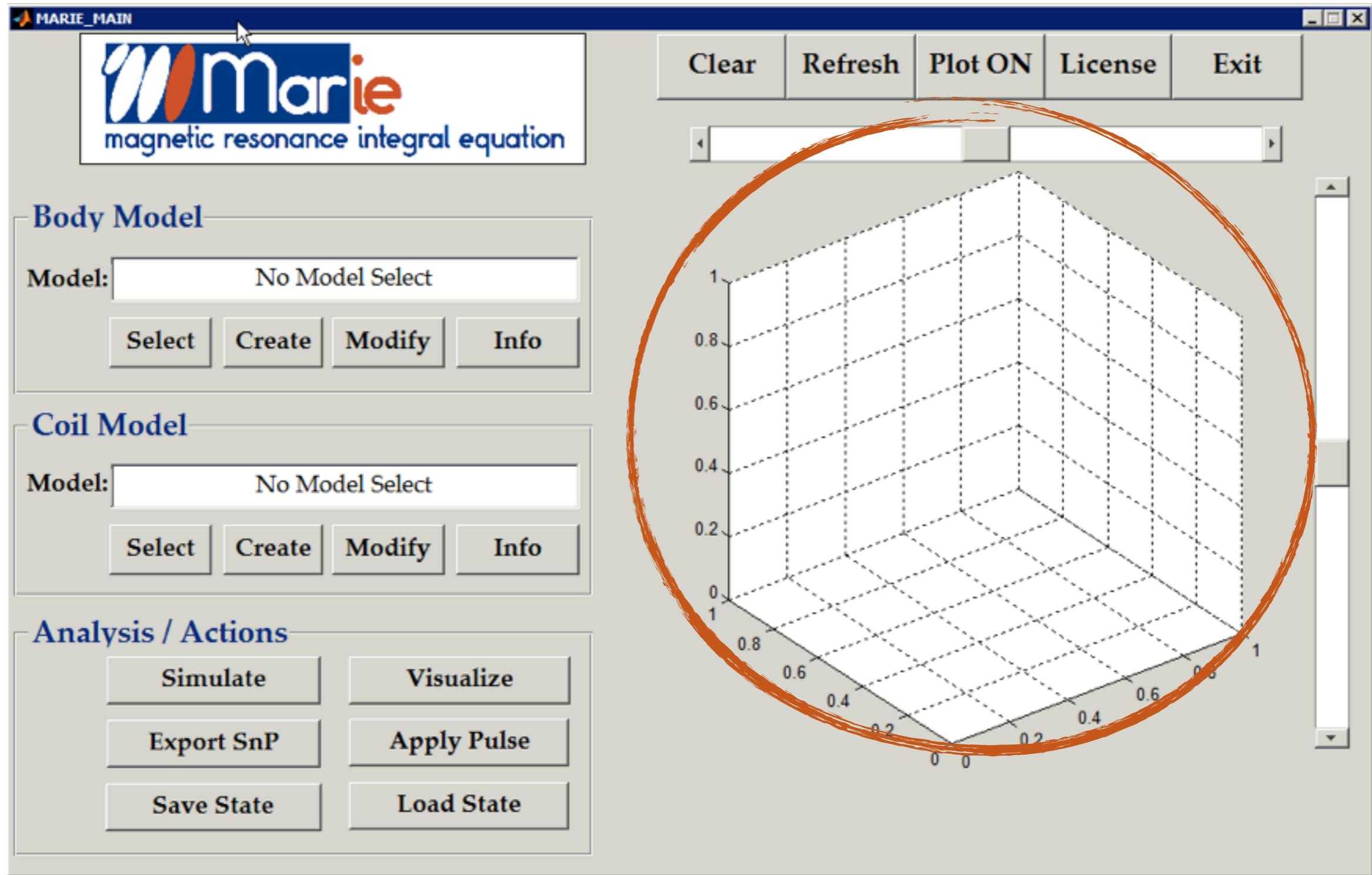
- ◆ Start the GUI
  - ◆ run **marie.m** script to call the GUI



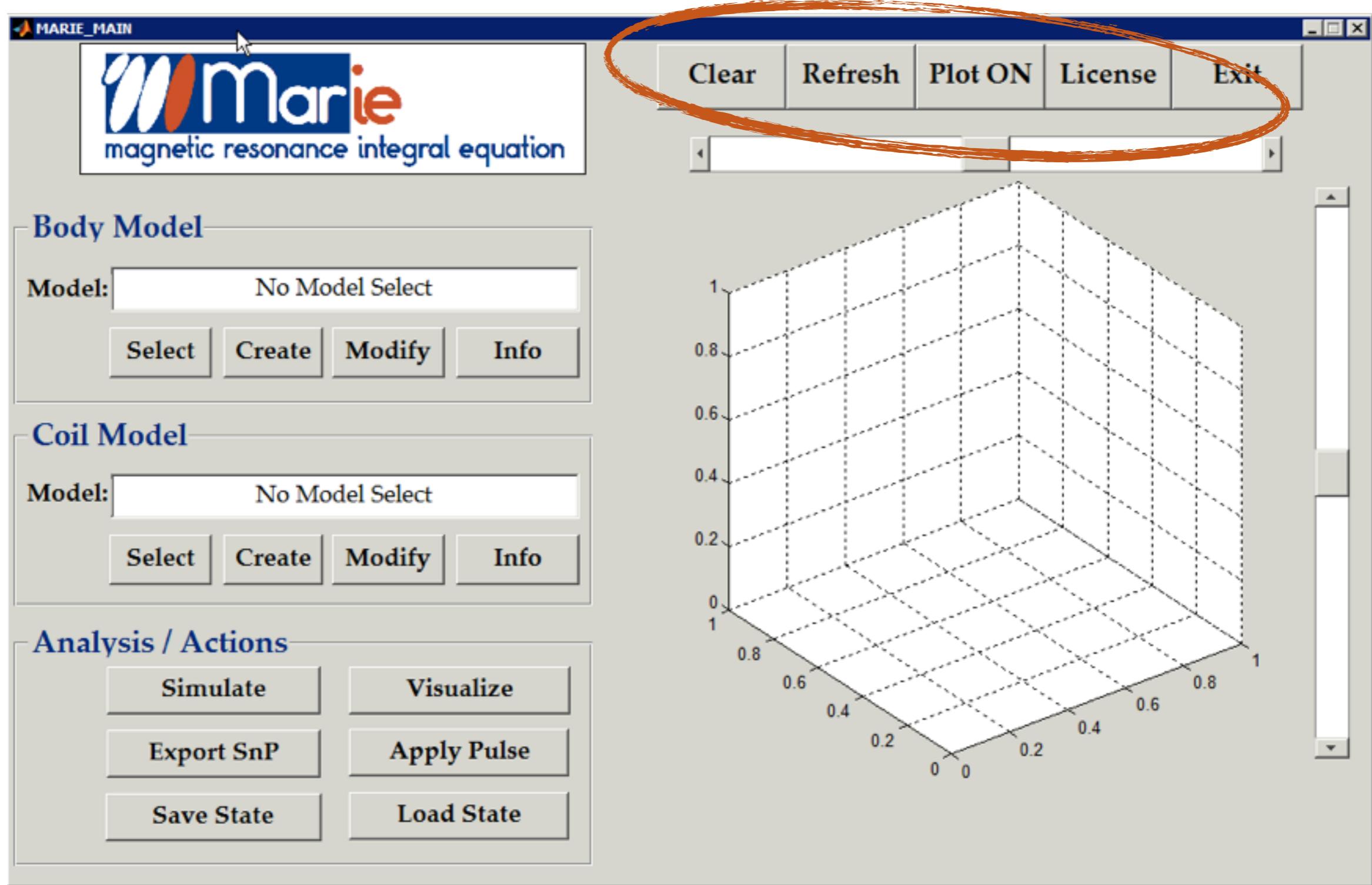
- ◆ Start the GUI
  - ◆ run **MARIE.m** script to call the GUI



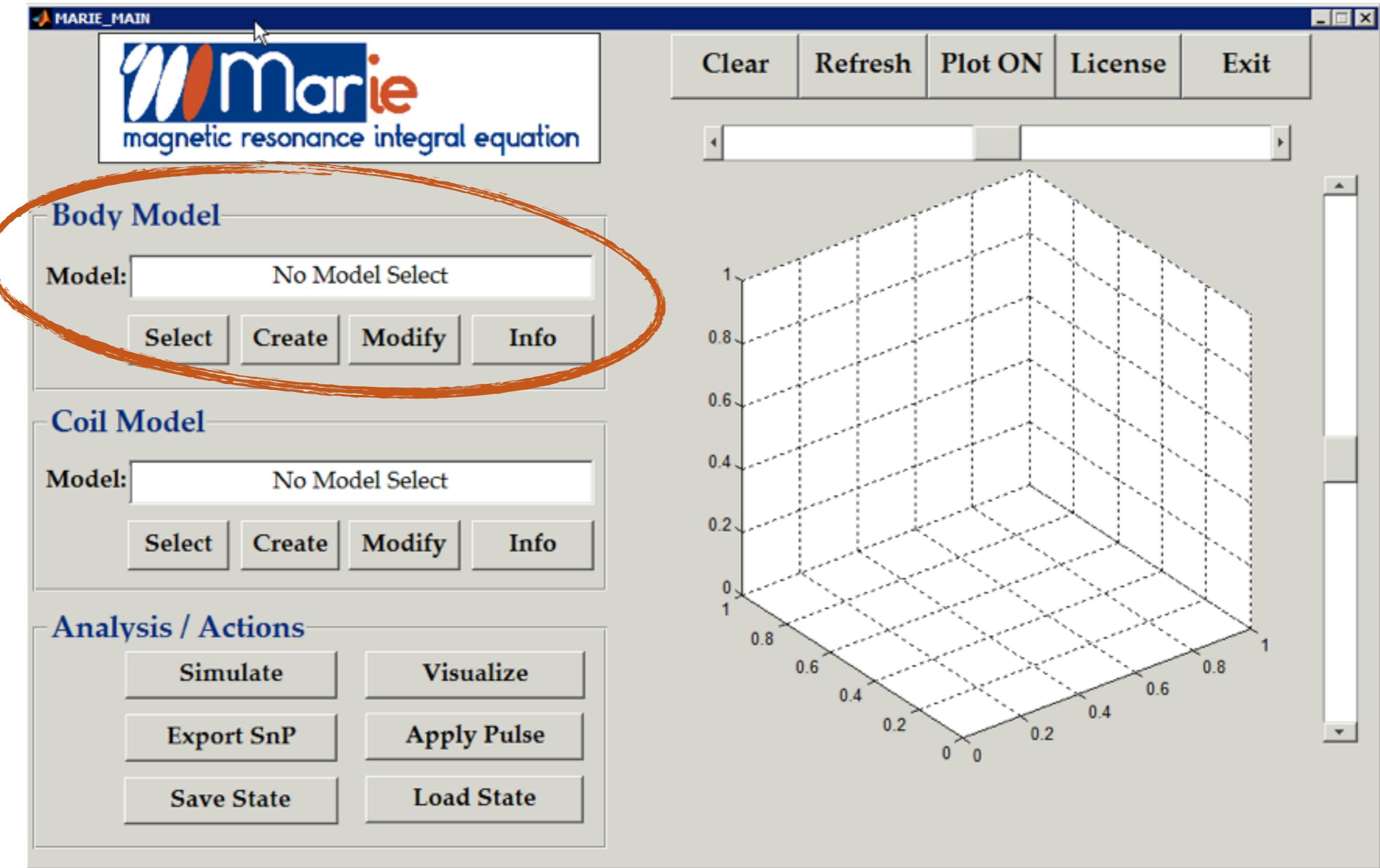
- ◆ Main parts of the GUI
  - ◆ figure of the loaded models, with elevation and azimuth controls



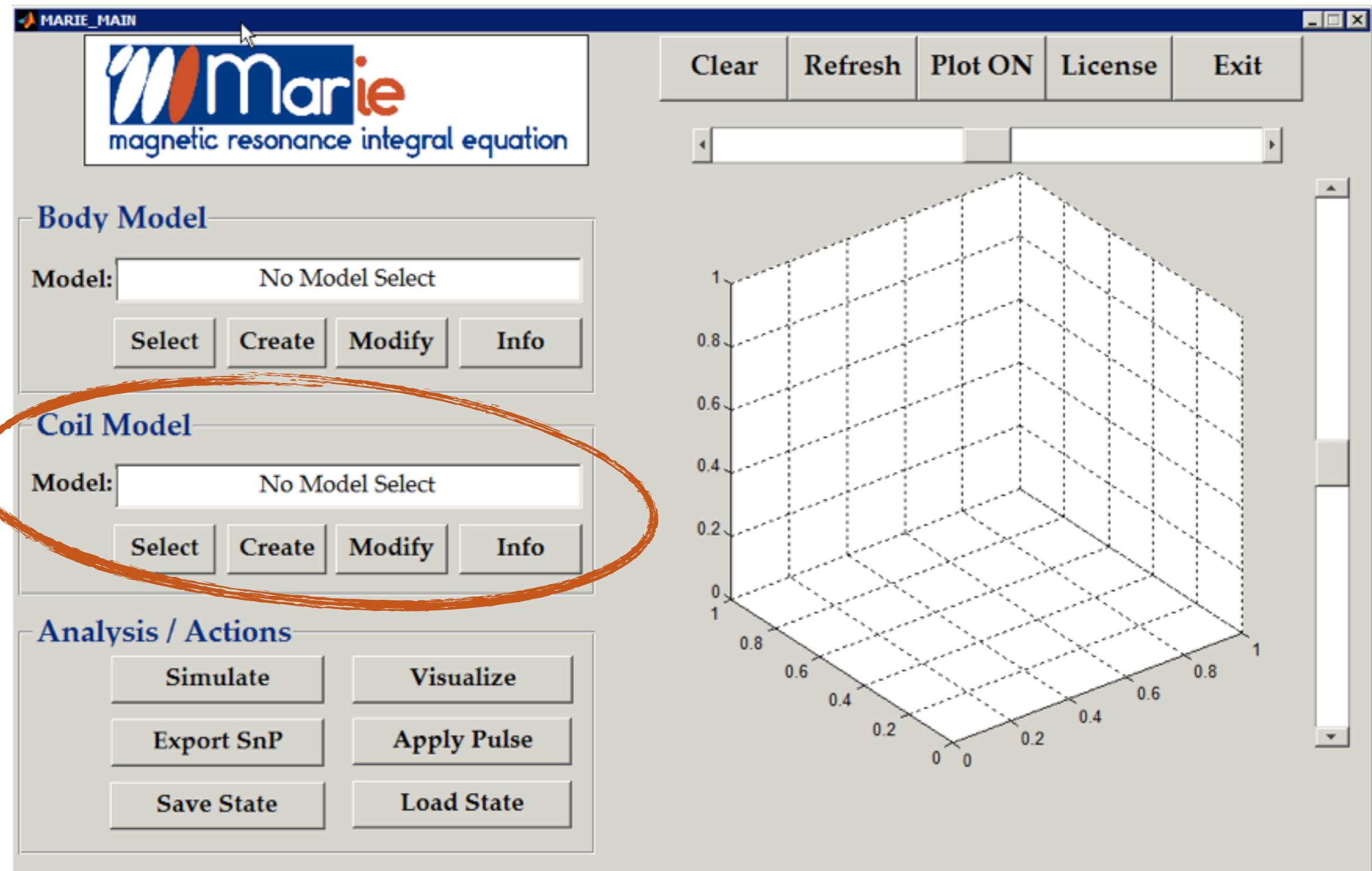
- ◆ Main parts of the GUI
  - ◆ miscellaneous functions



- ◆ Main parts of the GUI
  - ◆ function box related to the body model

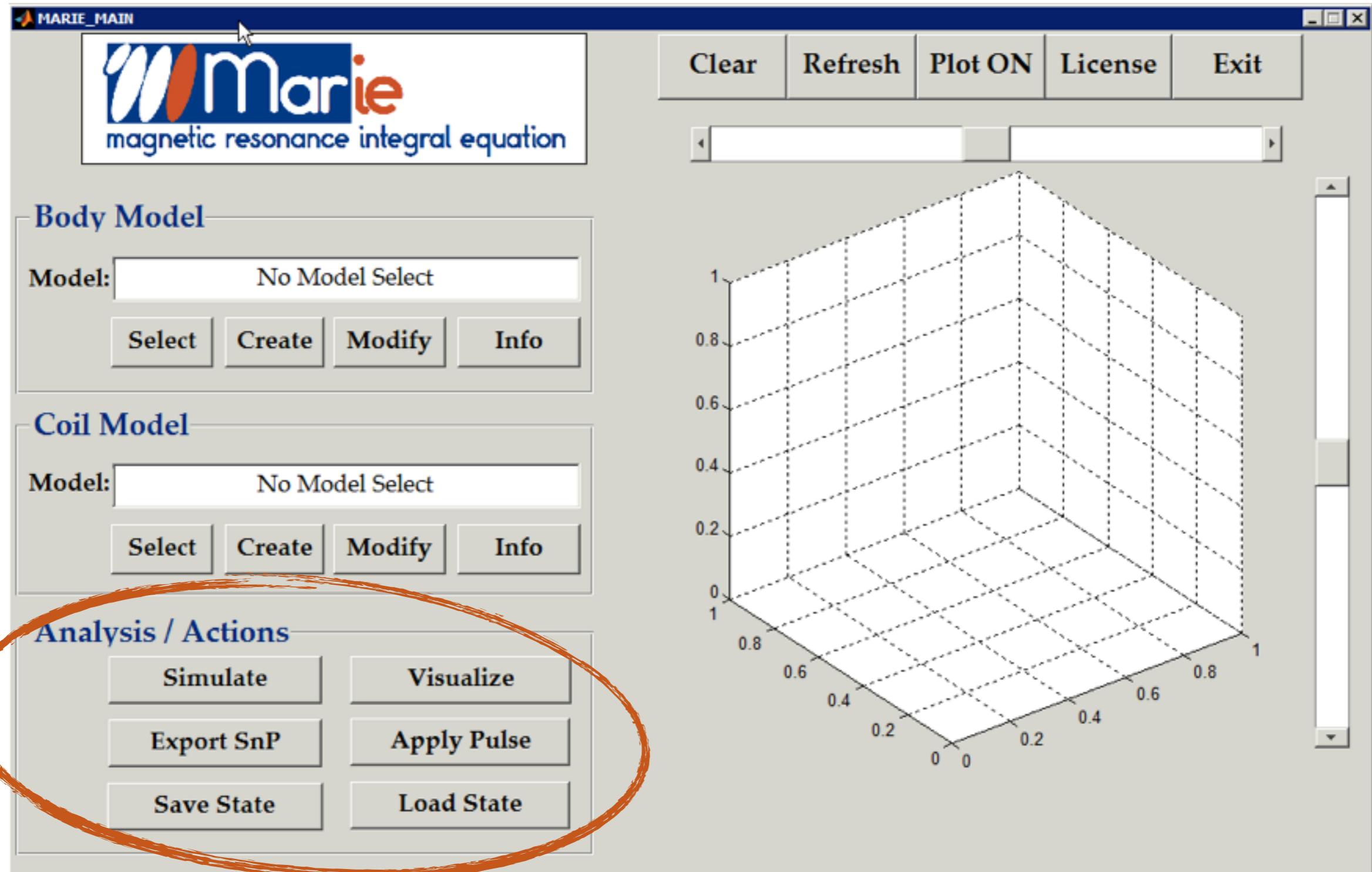


- ◆ Main parts of the GUI
  - ◆ function box related to the coil model



- ◆ Main parts of the GUI

- ◆ function box related to simulation, analysis and visualization



# file formats

## ◆ **File formats**

- ◆ MARIE uses several files to store information
  - ◆ info related to the Realistic Human Body Models
  - ◆ info related to the coil array geometry
  - ◆ info related to excitations and solutions of analysis
- ◆ For more information on body and coil models see
  - ◆ MARIE\_body.pdf
  - ◆ MARIE\_coil.pdf

## ◆ File formats: body model

- ◆ .vmm (volume marie model)
  - ◆ text file with body model information
  - ◆ includes names of other files with body data
    - ◆ one .mat file with body model data
    - ◆ four .dat files with voxel properties, and voxel coordinates

## ◆ File formats: coil model

- ◆ .smm (surface marie model)
  - ◆ text file with surface coil model information
  - ◆ includes name of other file with coil data
    - ◆ .msh file coil tessellation information
  - ◆ the .msh file is automatically generated from a .geo file by gmsh
    - ◆ .geo is a text file with 3D geometry description
- ◆ .wmm (wire marie model)
  - ◆ text file with wire coil model information
  - ◆ includes name of other file with coil data
    - ◆ .wsd (wire segment discretization), a simple segment description

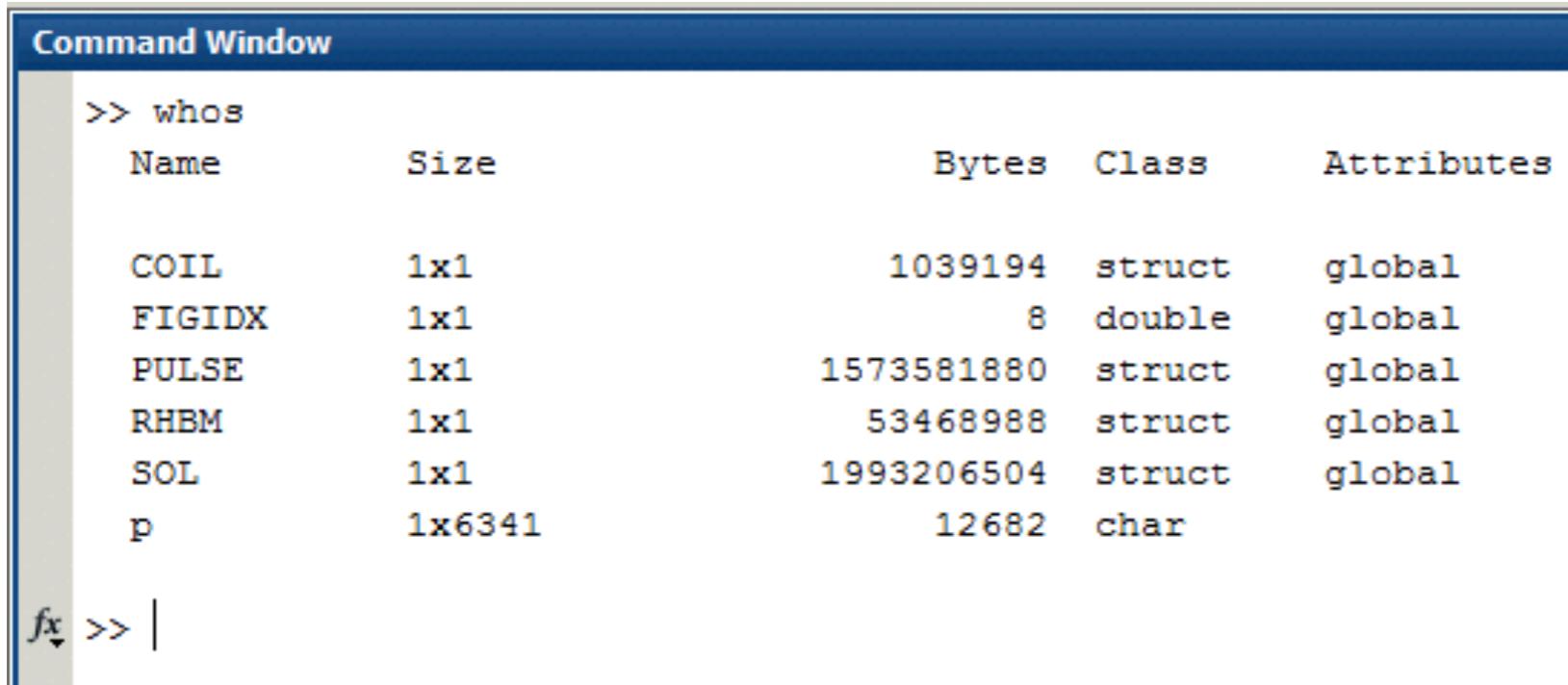
## ◆ File formats: solution and excitations

- ◆ .mat files are used to store and load states
  - ◆ at any point the main structures can be saved/loaded
  - ◆ useful to resume work
- ◆ .snp files are generated from frequency analysis
  - ◆ standard touchstone files with S, Y or Z port parameters
- ◆ .mps (marie pulse sequence)
  - ◆ a text file with voltage/current excitations for a pulse sequence
  - ◆ can be generated by circuit co-simulation
  - ◆ applied to a EM solution to obtain fields during a pulse sequence

# data structures

## ◆ Data structures

- ◆ **RHBM** - (realistic human body model) info related to the body
- ◆ **COIL** - info related to the coil array geometry
- ◆ **SOL** - structure that stores the solution of the analysis
- ◆ **PULSE** - structure that stores the solution of applying a pulse sequence



The image shows a MATLAB Command Window with the title "Command Window". The window displays the output of the command `>> whos`. The table lists six global variables: COIL, FIGIDX, PULSE, RHBM, SOL, and p. Each variable is a 1x1 struct with varying byte sizes.

| Name   | Size   | Bytes      | Class  | Attributes |
|--------|--------|------------|--------|------------|
| COIL   | 1x1    | 1039194    | struct | global     |
| FIGIDX | 1x1    | 8          | double | global     |
| PULSE  | 1x1    | 1573581880 | struct | global     |
| RHBM   | 1x1    | 53468988   | struct | global     |
| SOL    | 1x1    | 1993206504 | struct | global     |
| p      | 1x6341 | 12682      | char   |            |

- ◆ These structures are defined as global variables
  - ◆ available at the command line at any time - user can play with them
  - ◆ created upon call to MARIE, filled later with corresponding data

## ◆ Body data structure

- ◆ **RHBM** (realistic human body model)
- ◆ has geometric information of the scatterer
  - ◆ can be an inhomogeneous realistic human body model
  - ◆ or a homogeneous phantom or element (e.g. sphere)
- ◆ the information is given as voxel based material properties
  - ◆ used by the Volume Integral Equation modeling
- ◆ may include some pre-computed numeric information
  - ◆ VIE-related data (circulants from last solve)
  - ◆ Magnetic Resonance Green Functions (MRGFs)
  - ◆ to speed-up analysis and simulations

## ◆ RHBM (body data structure)

```
Command Window

FIGIDX      1x1                      8  double   global
PULSE        1x1                      1760 struct   global
RHBM         1x1          1312290508 struct   global
SOL          1x1          157354104  struct   global
p            1x6341                  12682 char

>> RHBM

RHBM =

    name: 'DUDE Head & Shoulders, 5mm, 7T'
        r: [4-D double]
    epsilon_r: [103x103x103 double]
    sigma_e: [103x103x103 double]
        rho: [103x103x103 double]
    idxS: [126858x1 double]
    freqfN: 300000000
    freqfK: 300000000
        fN: [4-D double]
        fK: [4-D double]
    freqM: []
    Dcoord: []
        P: []
    Um: []
    Sm: []
    Vm: []
        X: []
    M: []

I
```

- ◆ **RHBM (body data structure, standard info)**

- .name** - name or description of the model
- .r** - coordinates of the 3D grid defining the domain
- .epsilon\_r** - permittivity of each voxel of the grid
- .sigma\_e** - conductivity of each voxel of the grid
- .rho** - density of each voxel of the grid
- .idxS** - index of the elements of the grid that are not air (body voxels)

- ◆ this information is used by the Volume Integral Equation modeling

- ◆ **RHBM (body data structure, numeric data for VIE)**

- .fN / .freqfN - N operator and working frequency (at which was generated)

- .fK / .freqfK - K operator and working frequency (at which was generated)

- ◆ if the new computations are at the same frequency
  - ◆ fN and fK are re-used, and do not need to be re-generated

- ◆ **RHBM (body data structure, numeric MRGF data)**

- .freqM** - frequency at which the MRGF was generated
- .Dcoord** - coordinates of the DEIM interpolation points
- .P** - incidence matrix of the DEIM interpolation points
- .Um** - coil perturbation MRGF factor
- .Sm** - coil perturbation MRGF factor
- .Vm** - coil perturbation MRGF factor
- .X** - MRGF weighting matrix
- .M** - MRGF pre-computed solution

- ◆ this information is used by the accelerated Integral Equation solvers

## ♦ COIL data structure

- ◆ has geometric information of the coil array
- ◆ two different kind of coil models are used by MARIE
  
- ◆ surface coil models
  - ◆ planar metal structures
  - ◆ typical TX coil arrays
  - ◆ can include shield
  - ◆ modeled by Surface Integral Equation approach
  
- ◆ wire coil models
  - ◆ wire antennas
  - ◆ typical RX arrays are usually wire loops
  - ◆ modeled by Wire Integral Equation approach

## ◆ COIL data structure

```
Command Window

>> whos
  Name      Size            Bytes  Class     Attributes
  COIL      1x1              386   struct   global
  FIGIDX    1x1               8   double   global
  PULSE     1x1             1760   struct   global
  RHBM      1x1             1090   struct   global
  SOL       1x1             1408   struct   global
  p         1x6341           12682  char

>> COIL
COIL =
  name: 'No Model Selected'
  type: []

fx >> |
```

## ♦ COIL structure fields (common to both cases)

**.name** - string with name or description of the coil

**.type** - character that identifies the type of coil

- ◆ 'N' if no coil is loaded yet
- ◆ 'S' for surface coil model
- ◆ 'W' for wire coil model

**.Rhocoil** - resistivity of the coil material

- ◆ 0 indicates perfect electric conductor (PEC)
- ◆ note: coil losses in the surface coil model are not yet included

## ◆ COIL data structure: surface coils

```
Command Window
COIL          1x1           2223924  struct   global
FIGIDX        1x1             8  double   global
PULSE          1x1            1760  struct   global
RHBM           1x1            1090  struct   global
SOL            1x1            1408  struct   global
p              1x6341         12682  char

>> COIL

COIL =
    name: 'Circular Loop, R 40mm, Dist 120mm'
    type: 'S'
    Rhocoil: 1.257900000000000e-08
    Thickness: 5.000000000000000e-04
    index: [840x1 double]
    etod: [3x512 double]
    node: [3x330 double]
    edge: [2x840 double]
    elem: [4x512 double]
    index_elem: [1x1 struct]
        Ct: [3x512 double]
        Ln: [3x512 double]
        Pn: [3x3x512 double]
    port: [1x2 struct]
    Pcoil: []
    Ncoil: []
    Dwire: []
```

- ◆ **COIL structure fields (for surface coil model)**

- .**thickness** - thickness of the planar coil

- .**index** - indexes of the tessellation

- .**etod** - orientation of the edges of the tessellation

- .**node** - nodes of the tessellation

- .**edge** - edges of the tessellation

- .**elem** - element of the tessellation

- .**index\_elem** - index of each element

- .**Ct** - baricenters of the triangles

- .**Ln** - length of the triangles

- .**Pn** - perpendiculars of the triangles

- .**port** - port information

- ◆ this information is used by the Surface Integral Equation modeling
  - ◆ for more information about coil modeling, see MARIE\_coil.pdf file

## ◆ COIL data structure: wire coils

```
Command Window
COIL          1x1                  7892  struct   global
FIGIDX        1x1                  8     double  global
PULSE          1x1                  1760  struct   global
RHBM           1x1                  1090  struct   global
SOL            1x1                  1408  struct   global
p              1x6341                12682 char

>> COIL

COIL =
    name: 'Circular Loop, R 40mm, Dist 120mm'
    type: 'W'
    Rhocoil: 1.257900000000000e-08
    Thickness: []
    index: []
    etod: []
    node: []
    edge: []
    elem: []
    index_elem: []
        Ct: []
        Ln: []
        Pn: []
    Pcoil: [100x3 double]
    Ncoil: [100x3 double]
    Dwire: 5.000000000000000e-04
    port: [2x1 double]

fx >>
```

- ◆ **COIL structure fields (for wire coil model)**

- .Pcoil - positive edges of segments

- .Ncoil - negative edges of segments

- .Dwire - diameter of the wire

- .port - port information

- ◆ this information is used by the Wire Integral Equation modeling
- ◆ for more information about coil modeling, see MARIE\_coil.pdf file

## ◆ **SOL data structure**

- ◆ has the solution to the last analysis or simulation run, including:
  - ◆ 3D vector fields and volumetric currents, and 3D maps (SAR)
  - ◆ coil port parameters and coil current coefficients
  - ◆ absorbed power and globar SAR, and frequencies analyzed
- ◆ note: the solution can be given for multiple frequencies and ports

- ◆ SOL data structure
  - ◆ values after the analysis

```
Command Window

>> whos
  Name      Size            Bytes  Class    Attributes
  COIL      1x1             1039194  struct   global
  FIGIDX    1x1              8     double  global
  PULSE     1x1             1573581880 struct   global
  RHBM      1x1             53468988 struct   global
  SOL       1x1             1993206504 struct   global
  p         1x6341           12682   char

>> SOL
SOL =
  Jsol: [6-D double]
  Esol: [6-D double]
  Bsol: [6-D double]
  Ssol: [5-D double]
  Gsar: [4x3 double]
  Pabs: [4x3 double]
  Zparam: [4x4x3 double]
  freq: [298000000 299000000 300000000]
  Jcoil: [364x4x3 double]

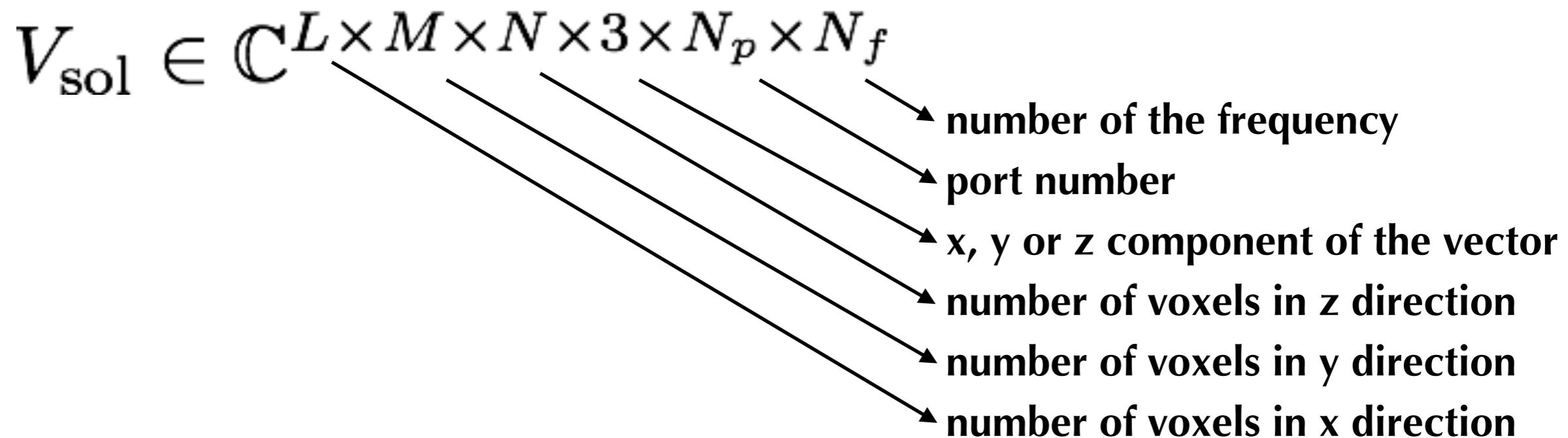
fx >> |
```

## ◆ **SOL (solution structure)**

- .Jsol** - volumetric current distribution in the body model
- .Esol** - electric field distribution
- .Bsol** - magnetic field distribution
- .Ssol** - voxel based SAR distribution
- .Gsar** - global SAR
- .Pabs** - absorbed power
- .Zparam** - coil port impedance parameters
- .Jcoil** - basis coefficients of the coil current distribution
- .freq** - frequencies

## ♦ SOL data structure

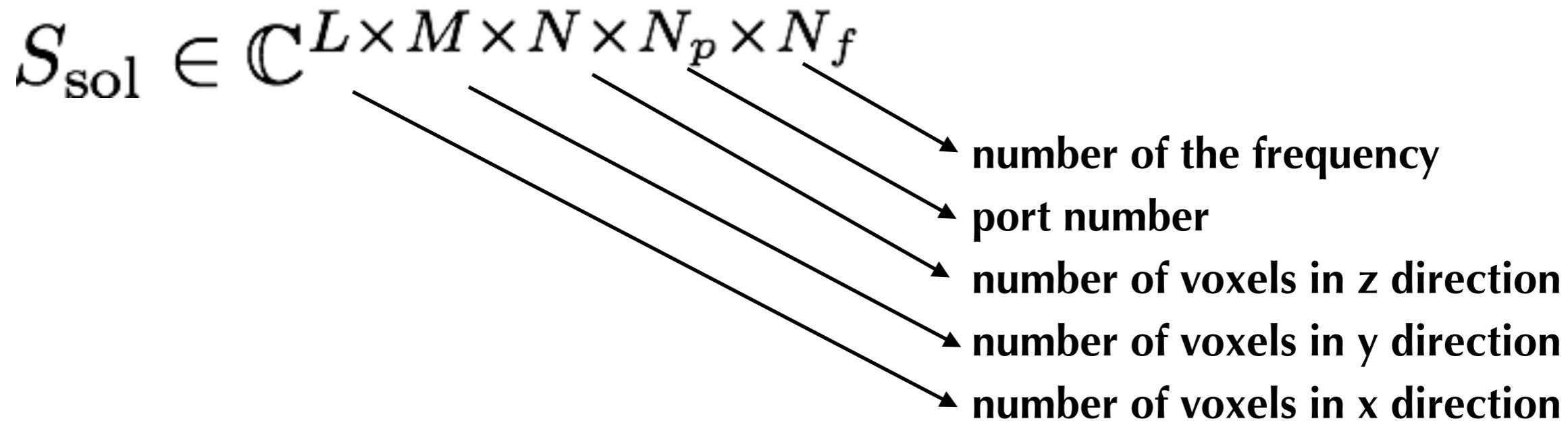
- ♦ 3D vectors format (Jsol, Esol, Bsol)



$V_{\text{sol}}(10, 20, 30, 2, 1, 5)$  is the y (2nd) component of V in the voxel:  
located in the 10-th x position  
20-th y position  
30-th z position of the grid  
due to the 1st port of the coil  
at the 5th frequency used in the analysis

- ♦ **SOL data structure**

- ♦ 3D maps format (Ssol)

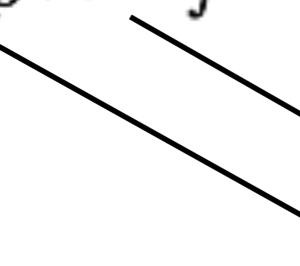


$S_{\text{sol}}(10, 20, 30, 1, 5)$  is the SAR value of the voxel:  
located in the 10-th x position  
20-th y position  
30-th z position of the grid  
due to the 1st port of the coil  
at the 5th frequency used in the analysis

- ◆ **SOL data structure**

- ◆ Quantities format (Gsar, Pabs)

$$Q_{\text{sol}} \in \mathbb{C}^{N_p \times N_f}$$

number of the frequency  
port number

$Q_{\text{sol}}(1, 5)$  is the quantity value  
due to the 1st port of the coil  
at the 5th frequency used in the analysis

- ◆ **SOL data structure**

- ◆ Coil current coefficient format ( $J_{coil}$ )

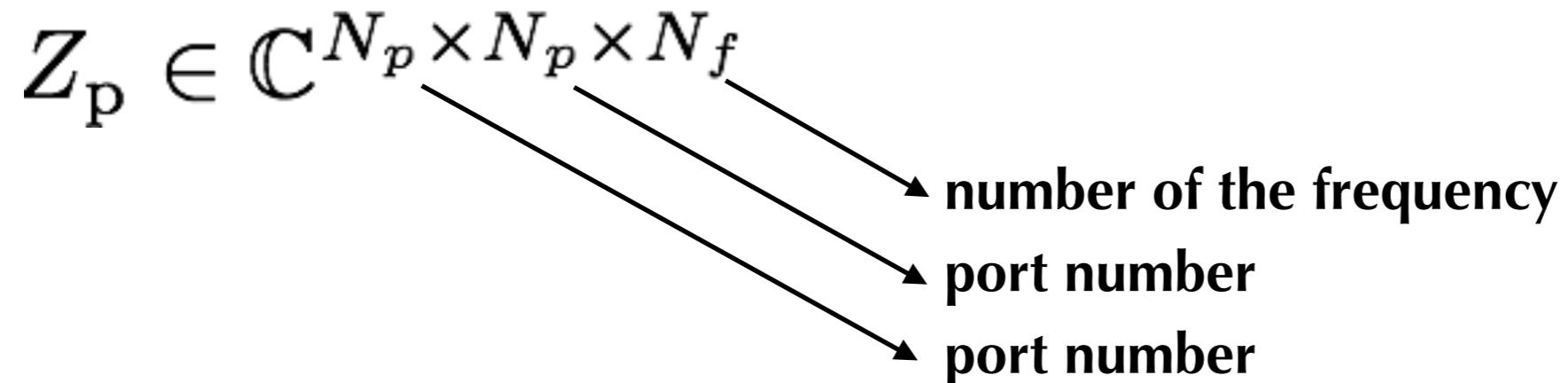
$$J_{coil} \in \mathbb{C}^{N_c \times N_p \times N_f}$$

- **number of the frequency**
    - **port number**
    - **number of coil current coefficient**

$J_{coil}(10, 1, 5)$  is the coil current coefficient of:  
the 10th basis function of the coil model  
due to the 1st port of the coil  
at the 5th frequency used in the analysis

## ♦ SOL data structure

### ♦ Port parameter format (Zparam)



$Z_p(2, 1, 5)$  is the Z parameter value:  
from the 1st port of the coil  
to the 2nd port of the coil  
at the 5th frequency used in the analysis

## ◆ PULSE data structure

- ◆ solution after applying a pulse sequence (from .mps) to the solution
  - ◆ e.g. after tuning and matching, the voltage/current values at the ports

```
Command Window

>> whos
  Name      Size            Bytes  Class    Attributes
  COIL      1x1           1039194  struct   global
  FIGIDX    1x1              8   double   global
  PULSE     1x1          2622635160  struct   global
  RHBM      1x1           53468988  struct   global
  SOL       1x1          1993206504  struct   global
  p         1x6341          12682   char

>> PULSE

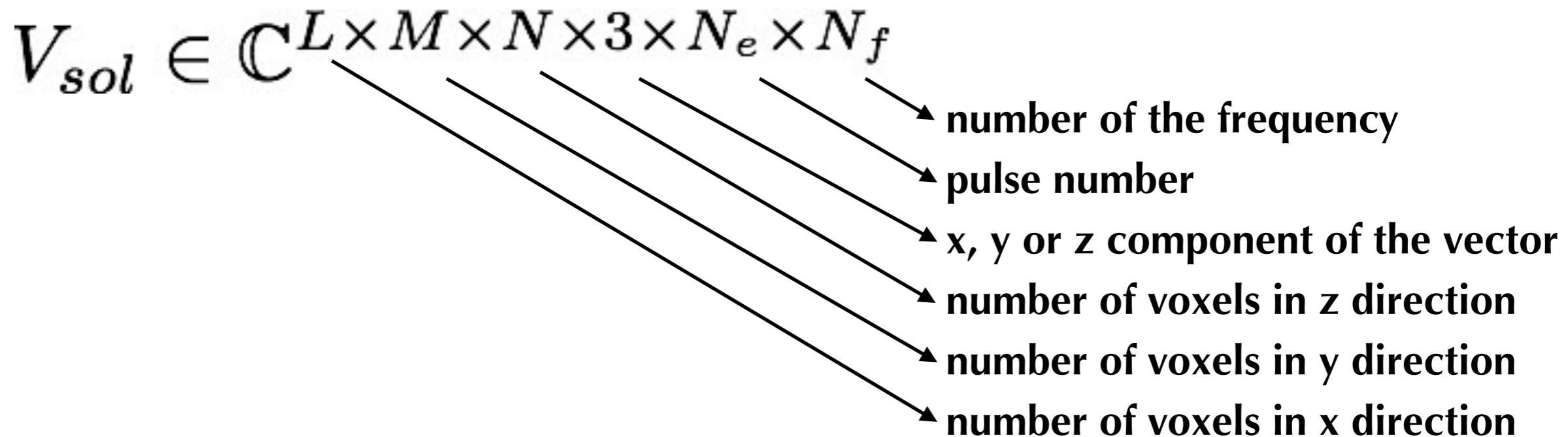
PULSE =
  Jcoil: [364x5x3 double]
  Jsol: [6-D double]
  Ssol: [5-D double]
  Esol: [6-D double]
  Bsol: [6-D double]
  Pabs: [5x3 double]
  Gsar: [5x3 double]
  step: [5x1 double]
  freq: [298000000 299000000 300000000]
  Ipulse: [4x5 double]
  Vpulse: [4x5 double]
```

## ◆ PULSE (solution structure)

- .Jcoil** - basis coefficients of the coil current distribution
- .Jsol** - volumetric current distribution in the body model
- .Esol** - electric field distribution
- .Bsol** - magnetic field distribution
- .Ssol** - voxel based SAR distribution
- .Gsar** - global SAR
- .Pabs** - absorbed power
- .Vpulse** - voltages at the ports at each time step
- .Ipulse** - currents at the ports at each time step
- .step** - array with time stamp of each pulse or pulse number
- .freq** - array with the frequencies

## ♦ SOL data structure

- ♦ 3D vectors format (Jsol, Esol, Bsol)



$V_{sol}(10, 20, 30, 2, 1, 5)$  is the y (2nd) component of V in the voxel:  
located in the 10-th x position  
20-th y position  
30-th z position of the grid  
due to the 1st pulse applied  
at the 5th frequency used in the analysis

## ♦ SOL data structure

- ♦ 3D maps format (Ssol)

$$S_{\text{sol}} \in \mathbb{C}^{L \times M \times N \times N_e \times N_f}$$

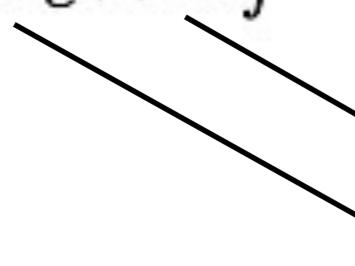
- number of the frequency
- pulse number
- number of voxels in z direction
- number of voxels in y direction
- number of voxels in x direction

$S_{\text{sol}}(10, 20, 30, 1, 5)$  is the SAR value of the voxel:  
located in the 10-th x position  
20-th y position  
30-th z position of the grid  
due to the 1st pulse applied  
at the 5th frequency used in the analysis

- ◆ **SOL data structure**

- ◆ Quantities format (Gsar, Pabs)

$$Q_{\text{sol}} \in \mathbb{C}^{N_e \times N_f}$$

 number of the frequency  
pulse number

$Q_{\text{sol}}(1, 5)$  is the quantity value  
due to the 1st pulse applied  
at the 5th frequency used in the analysis

- ◆ **SOL data structure**

- ◆ Coil current coefficient format ( $J_{coil}$ )

$$J_{coil} \in \mathbb{C}^{N_c \times N_e \times N_f}$$

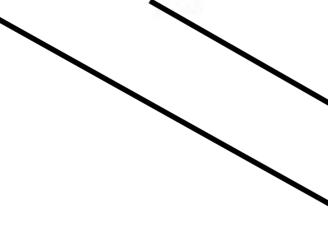
- **number of the frequency**
    - **pulse number**
    - **number of coil current coefficient**

$J_{coil}(10, 1, 5)$  is the coil current coefficient of:  
the 10th basis function of the coil model  
due to the 1st pulse applied  
at the 5th frequency used in the analysis

- ◆ **SOL data structure**

- ◆ Port excitation ( $V_{pulse}$ ,  $I_{pulse}$ )

$$P_p \in \mathbb{C}^{N_e \times N_p}$$

Two arrows point from the matrix  $P_p$  to the labels "port number" and "pulse excitation number".

**port number**  
**pulse excitation number**

$V_p(2, 3)$     **is the voltage:**  
                        **applied by 2nd pulse**  
                        **at the 3rd port of the coil**

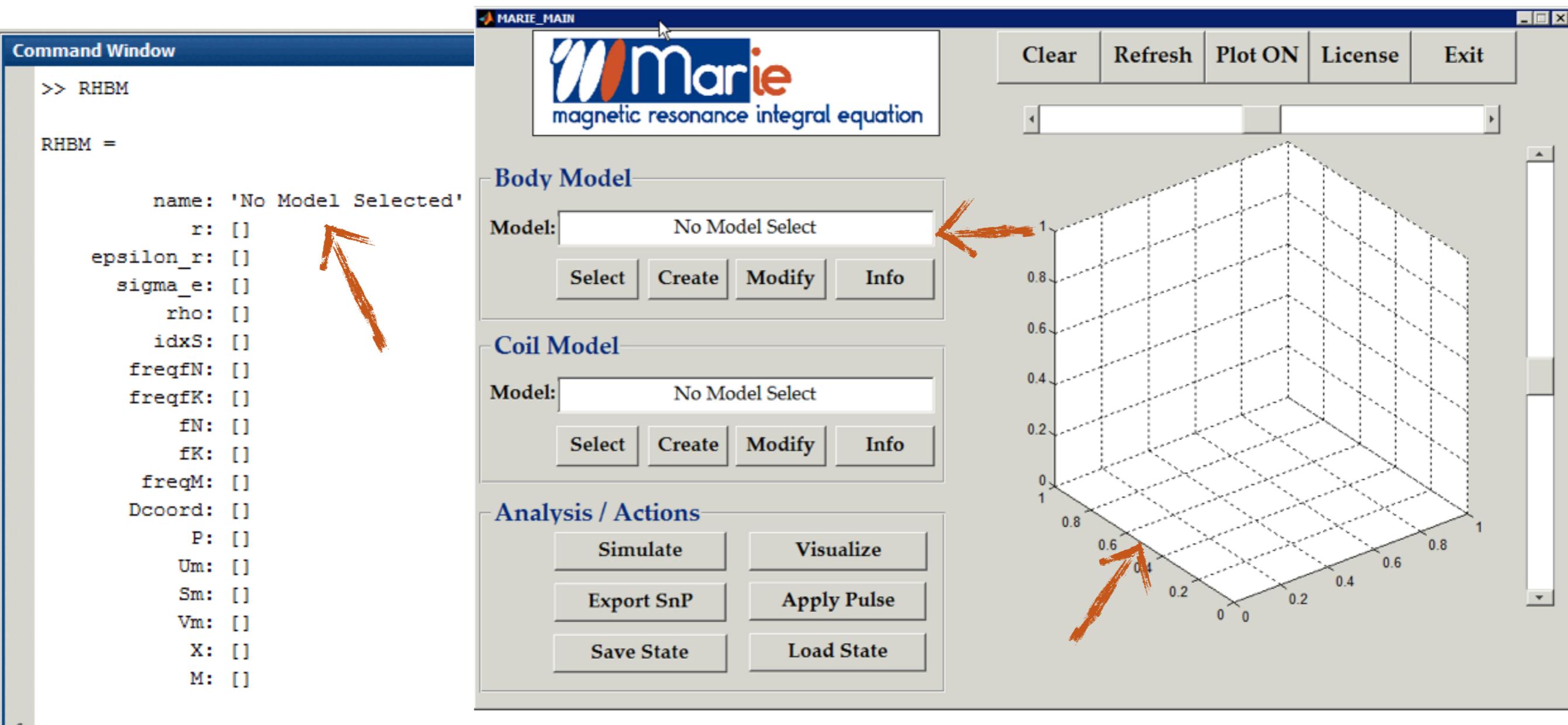
gui - body model box

## ◆ Forenote

- ◆ all quantities in the GUI are represented in SI units
  - ◆ in particular, dimensions are always in meters
- ◆ no reference to units are present in the GUI

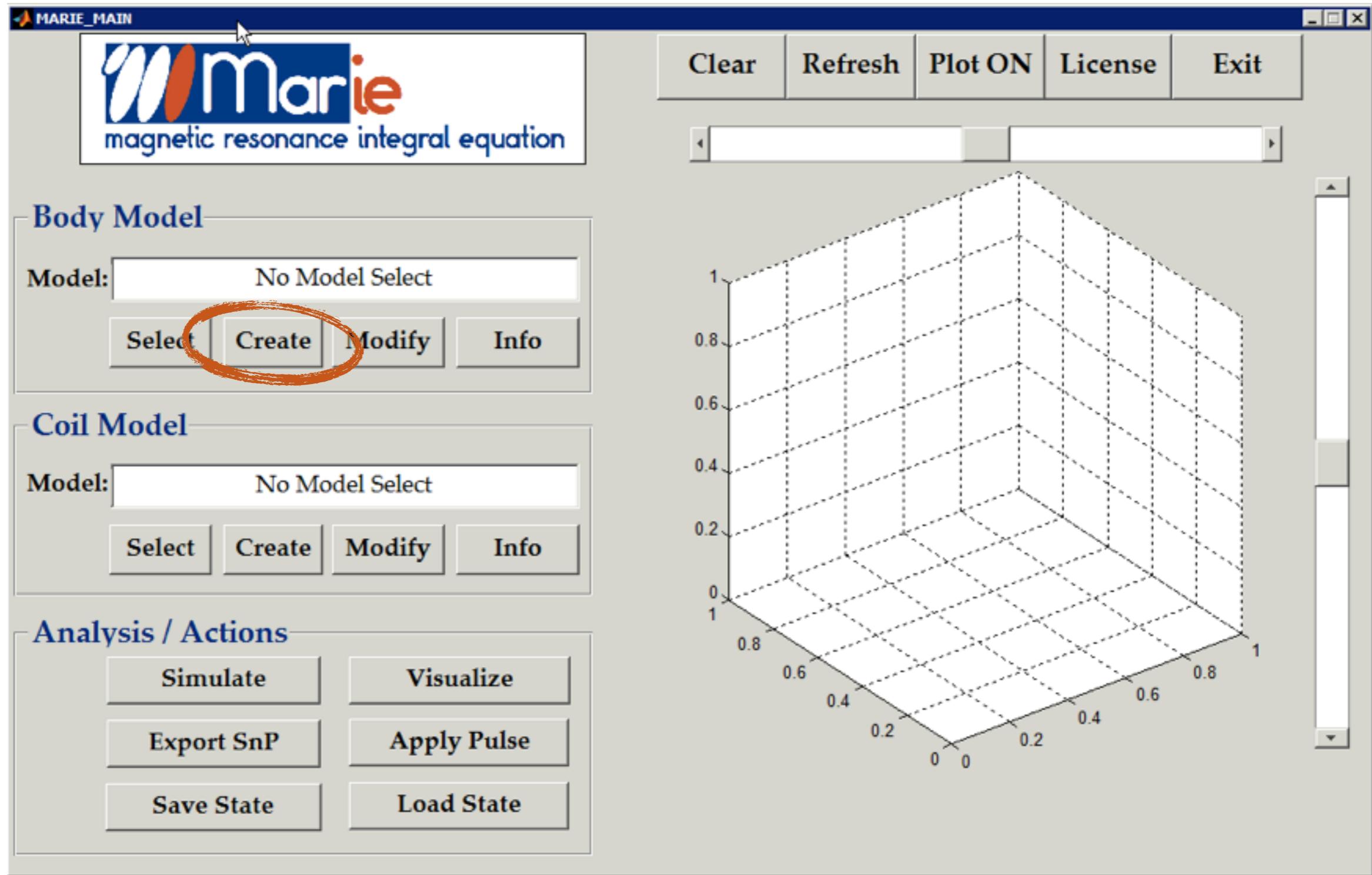
## ◆ Function box related to the body model

- ◆ the initial model is empty
- ◆ “No Model Selected” in the body model field, figure empty



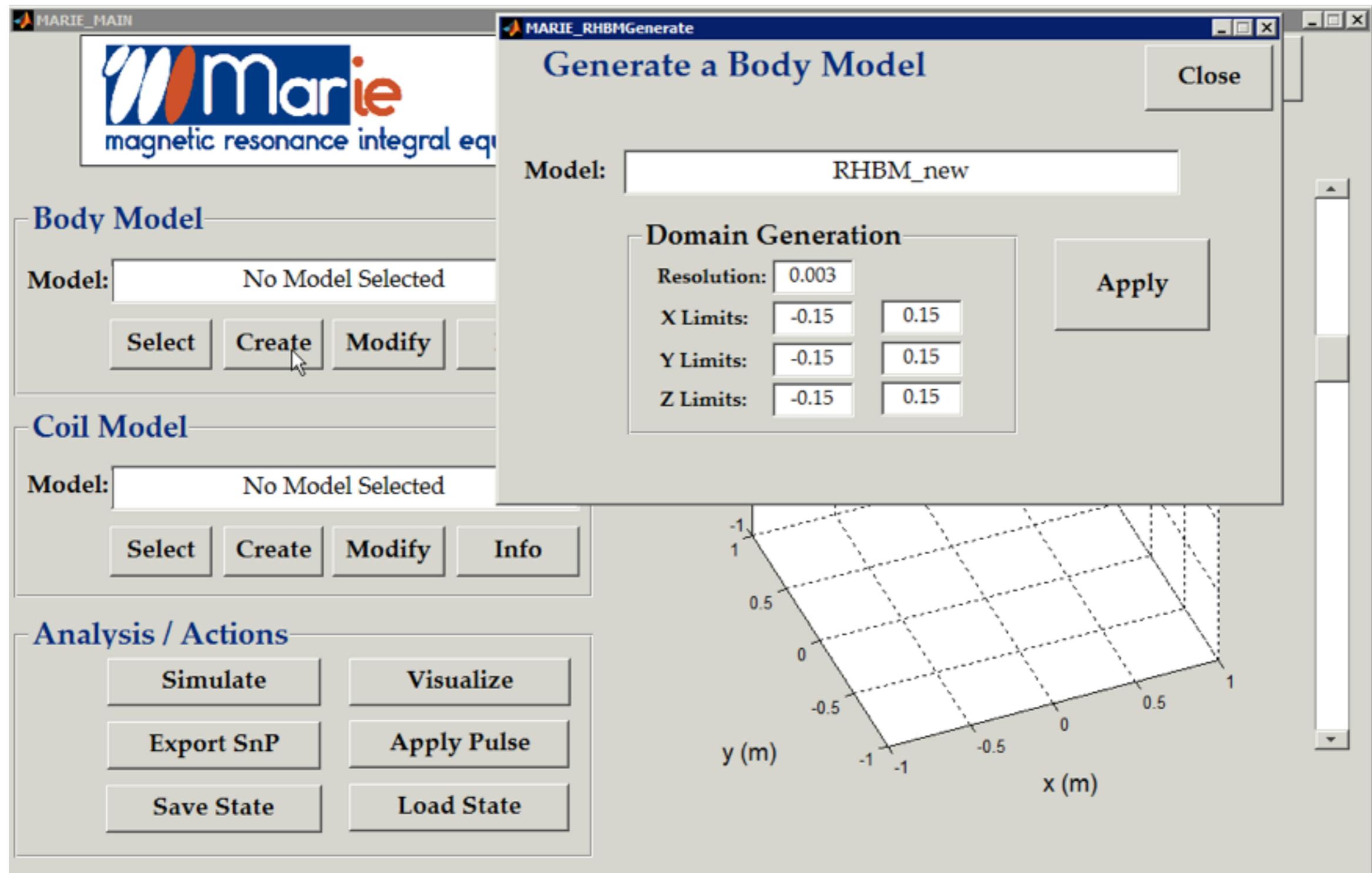
- ♦ Create a body model

- ♦ push “Create” to generate a volumetric model from scratch



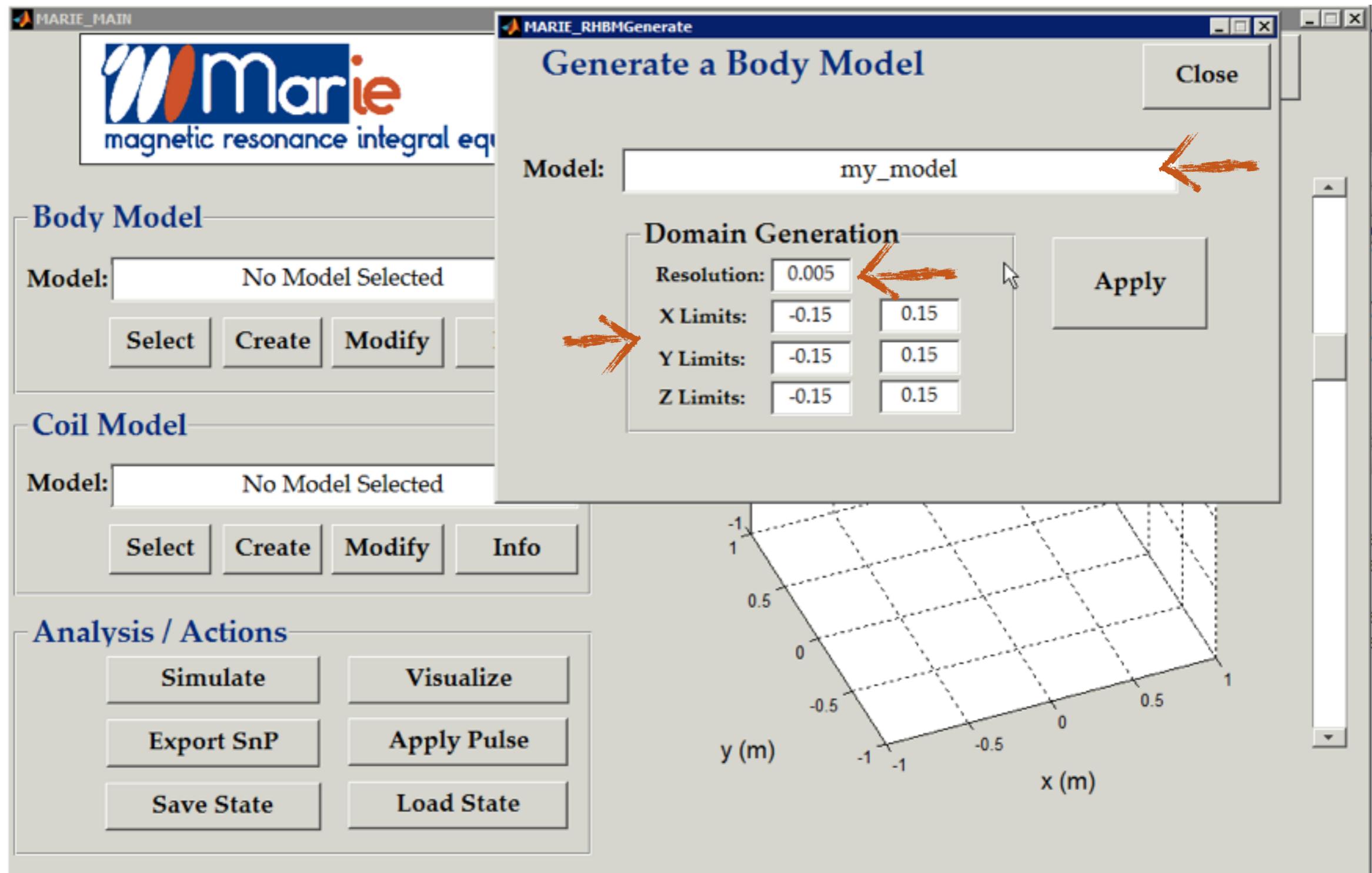
## ◆ Create a body model

- ◆ generate body model window will pop-up



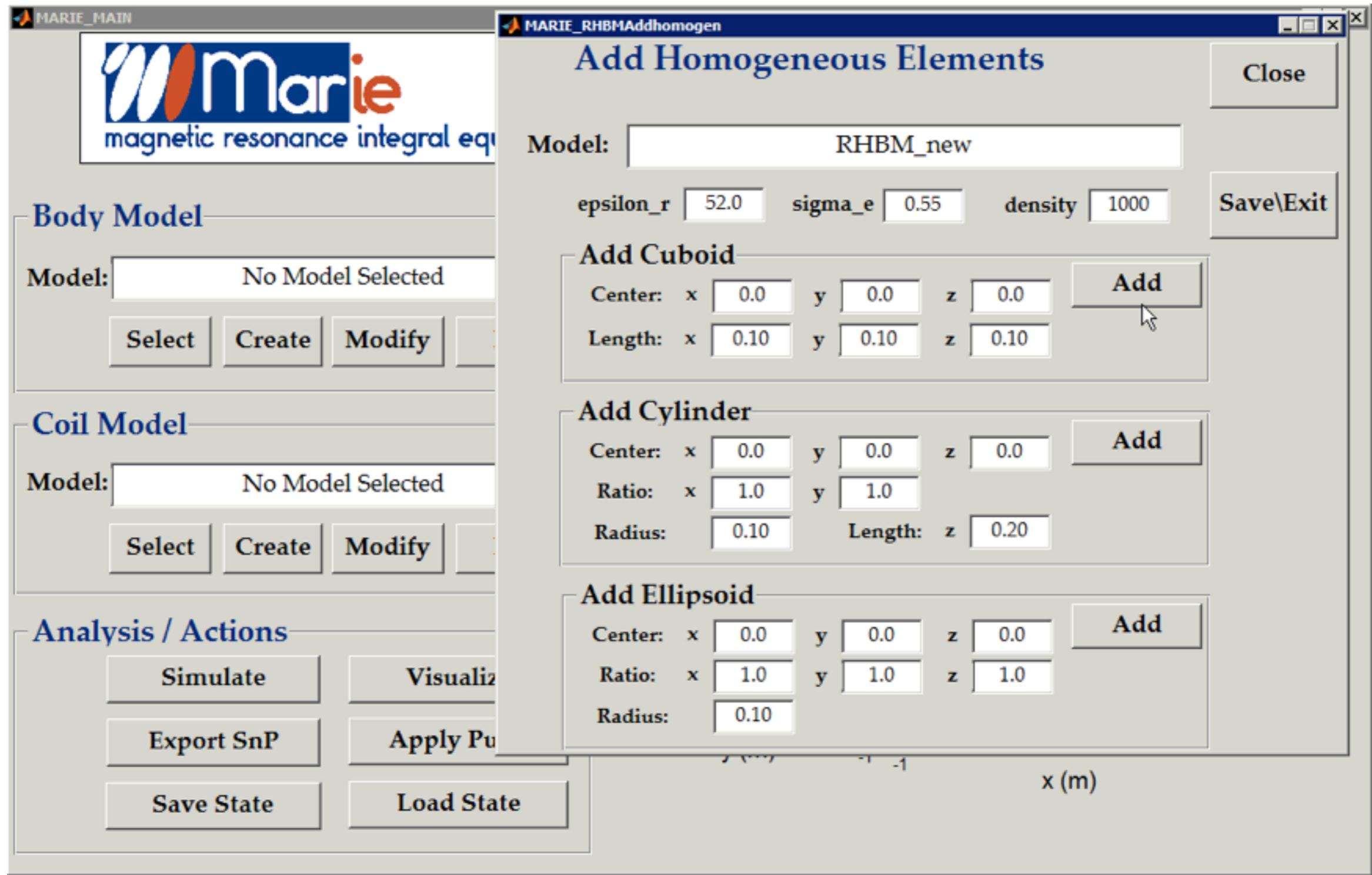
## ◆ Create a body model

- ◆ modify the size and resolution of the domain, and press apply



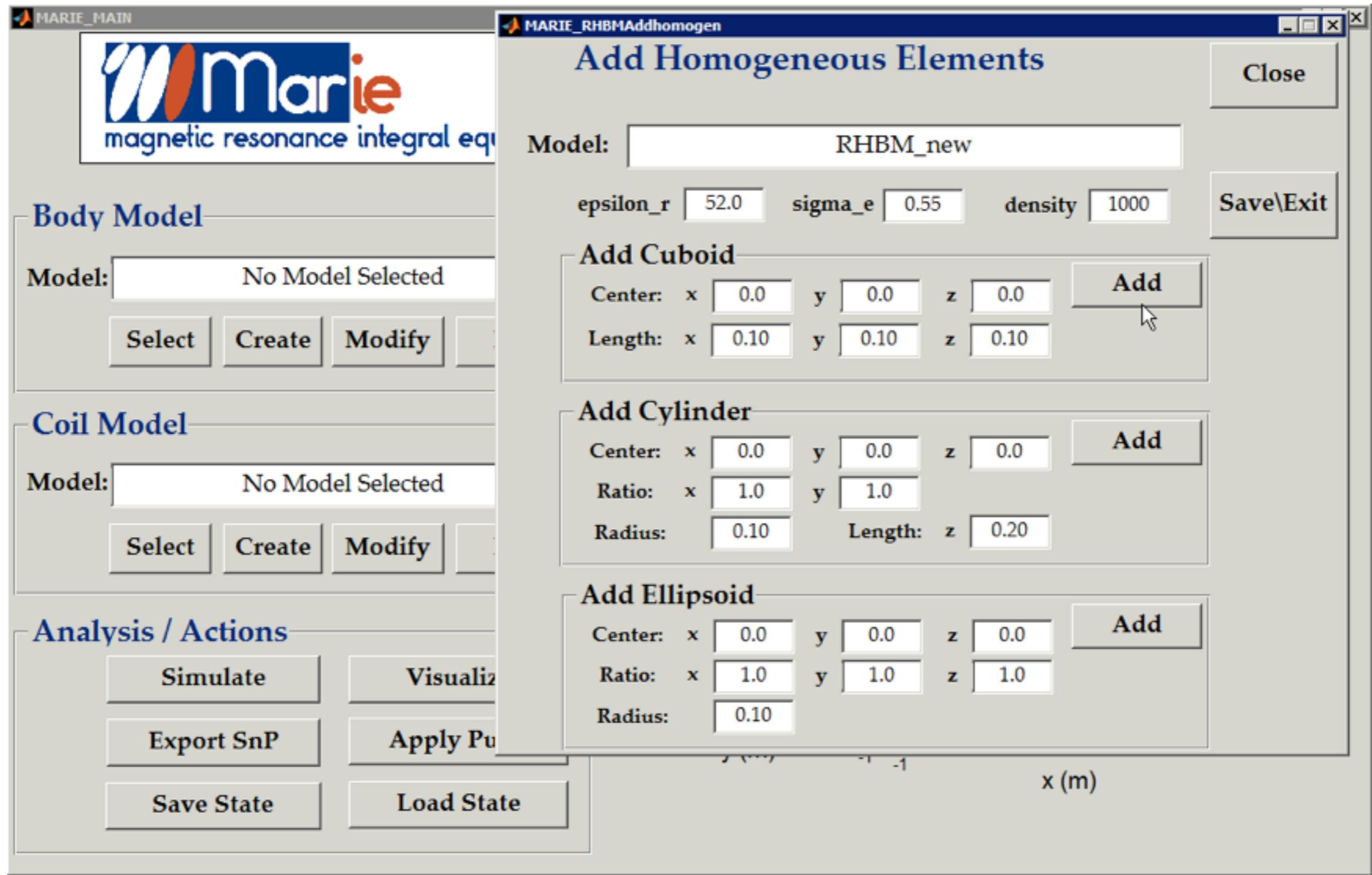
## ◆ Create a body model

- ◆ an interface to add homogeneous shapes will appear



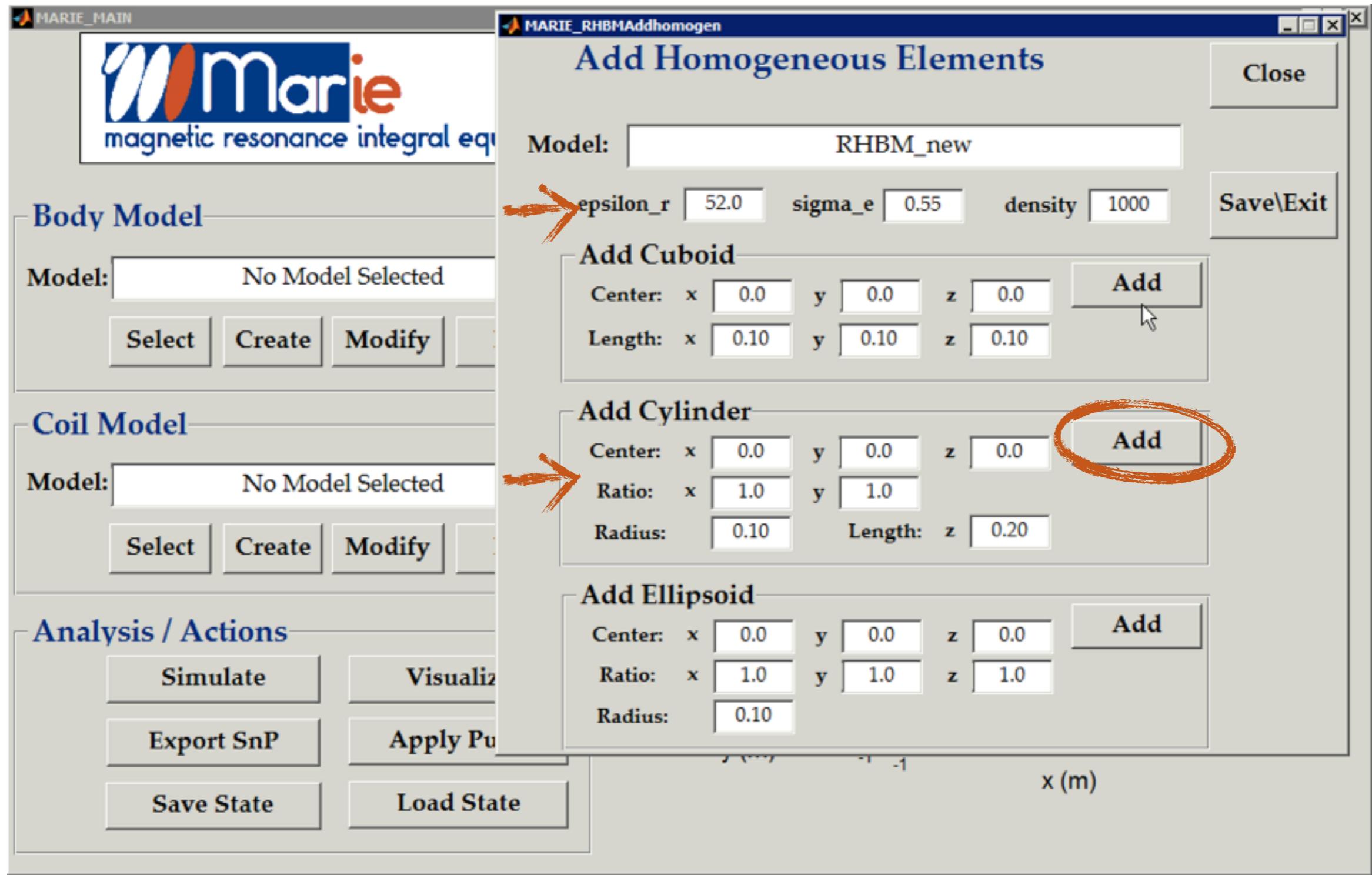
## ◆ Create a body model

- ◆ note: any part of the elements outside the domain will be truncated



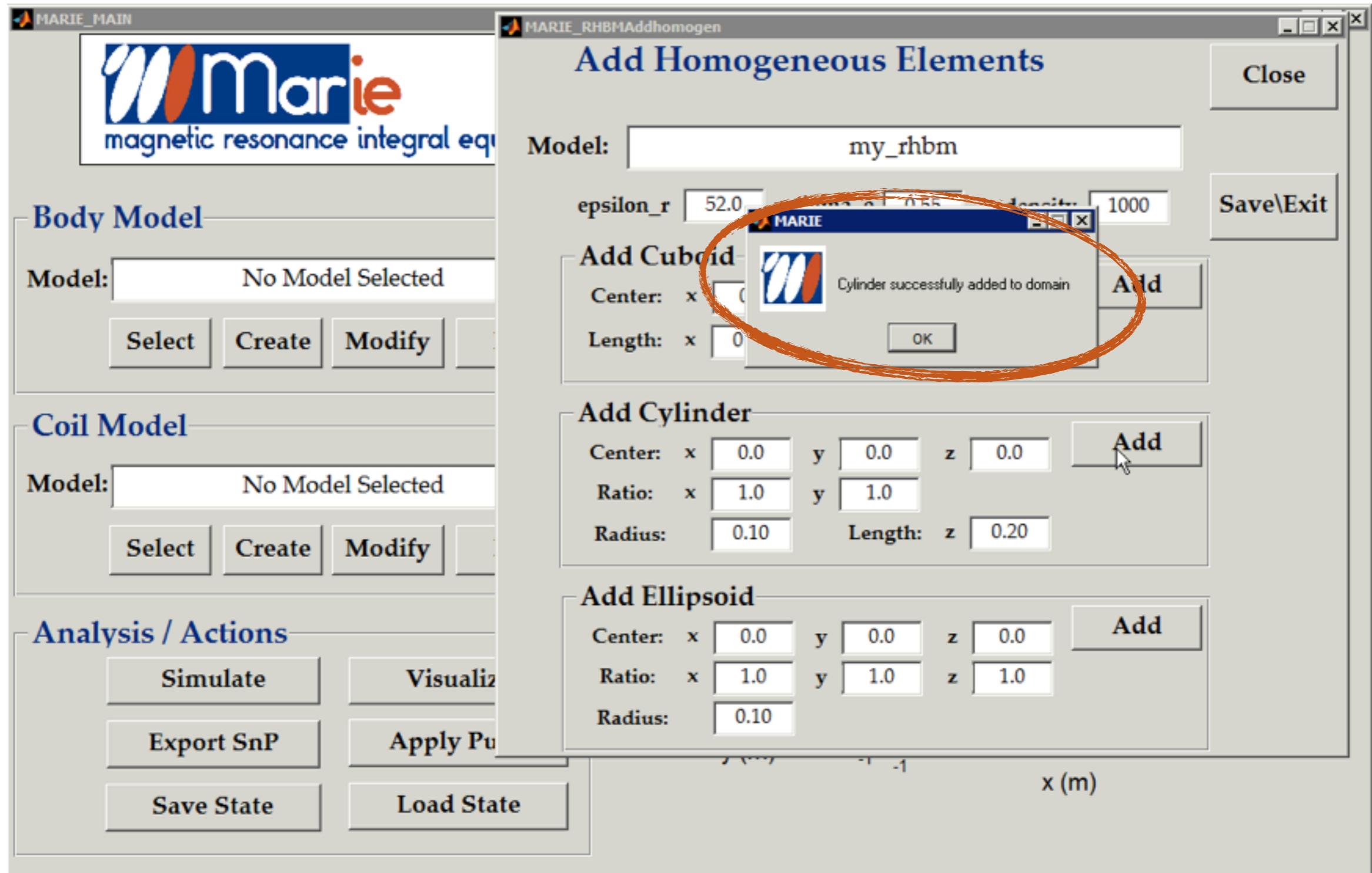
## ♦ Create a body model

- ♦ you can specify a given cylinder, and push “add” to add to the model



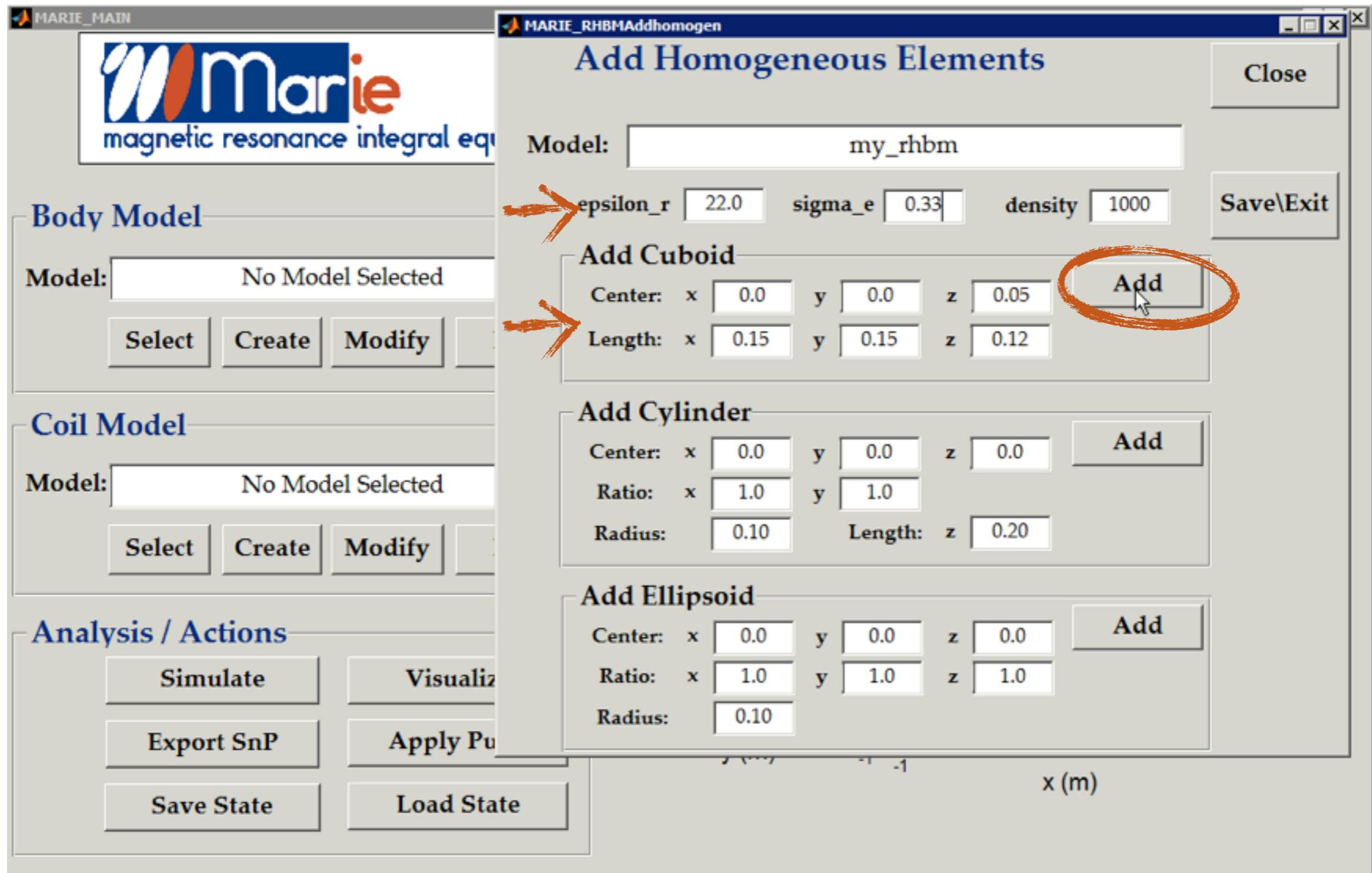
## ◆ Create a body model

- ◆ a window will indicate you have added the element



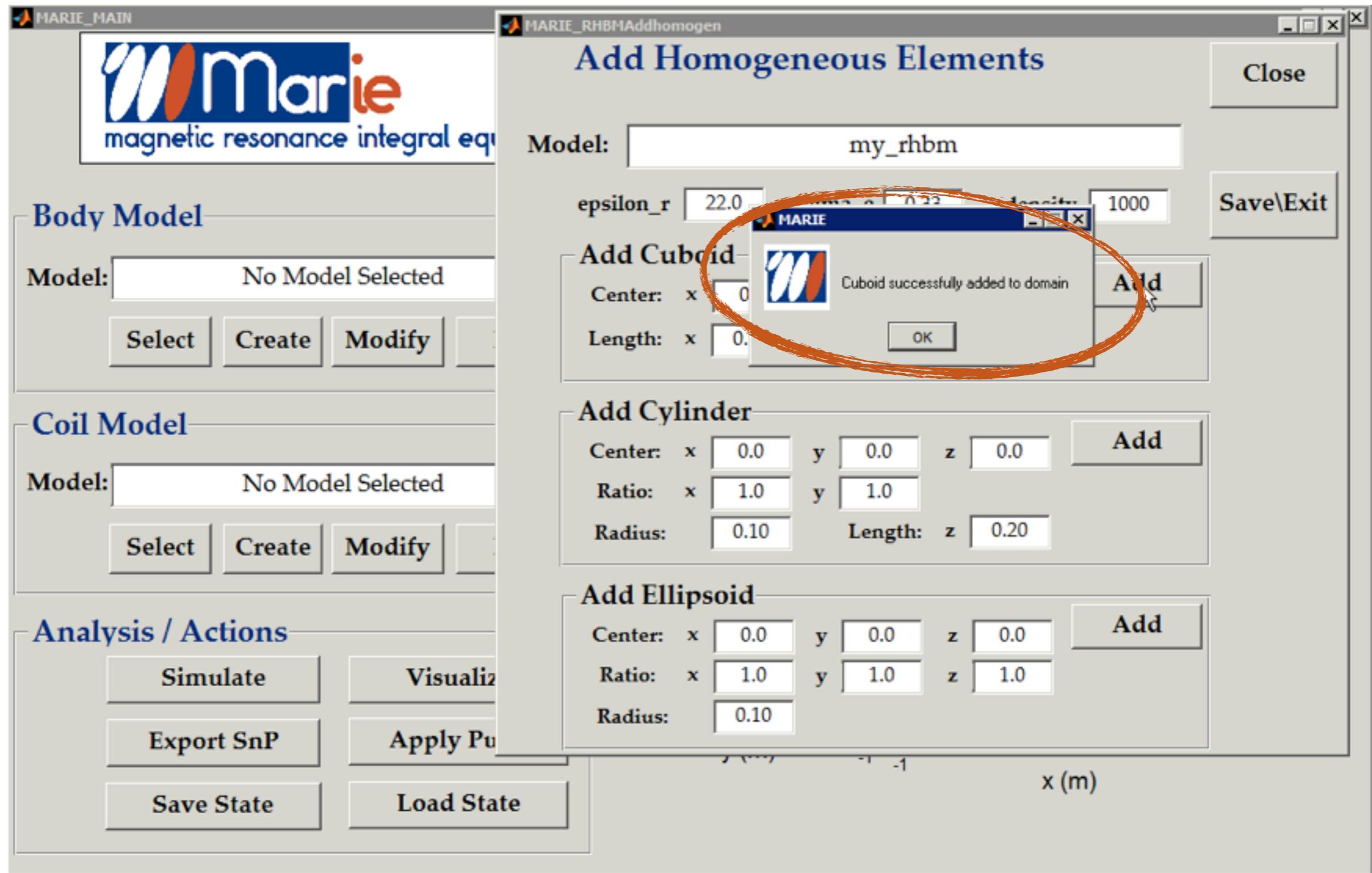
## ♦ Create a body model

- ♦ the same with a cuboid, pushing “add” to include it to the model

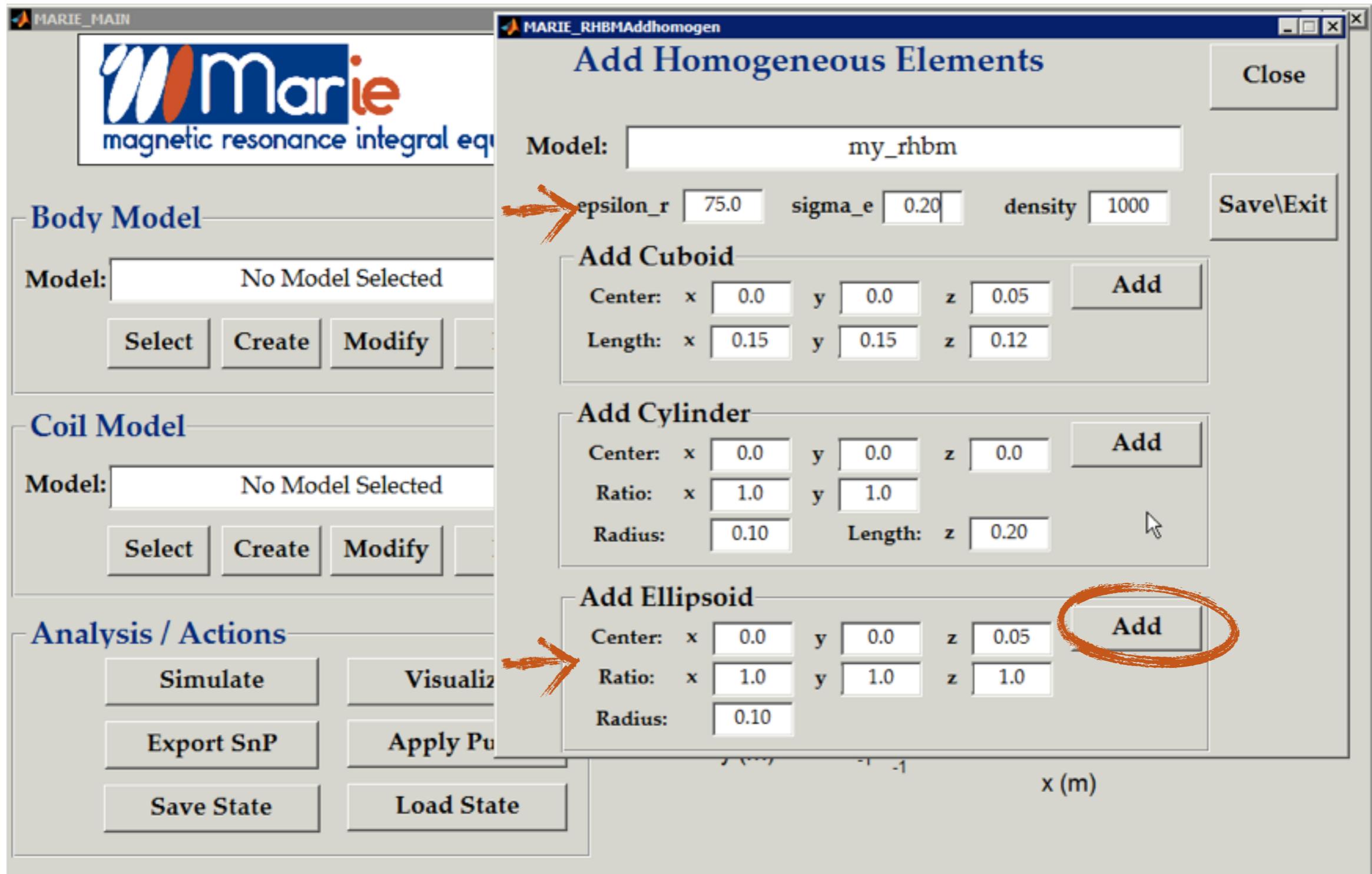


## ♦ Create a body model

- ♦ a window will indicate you have added the element

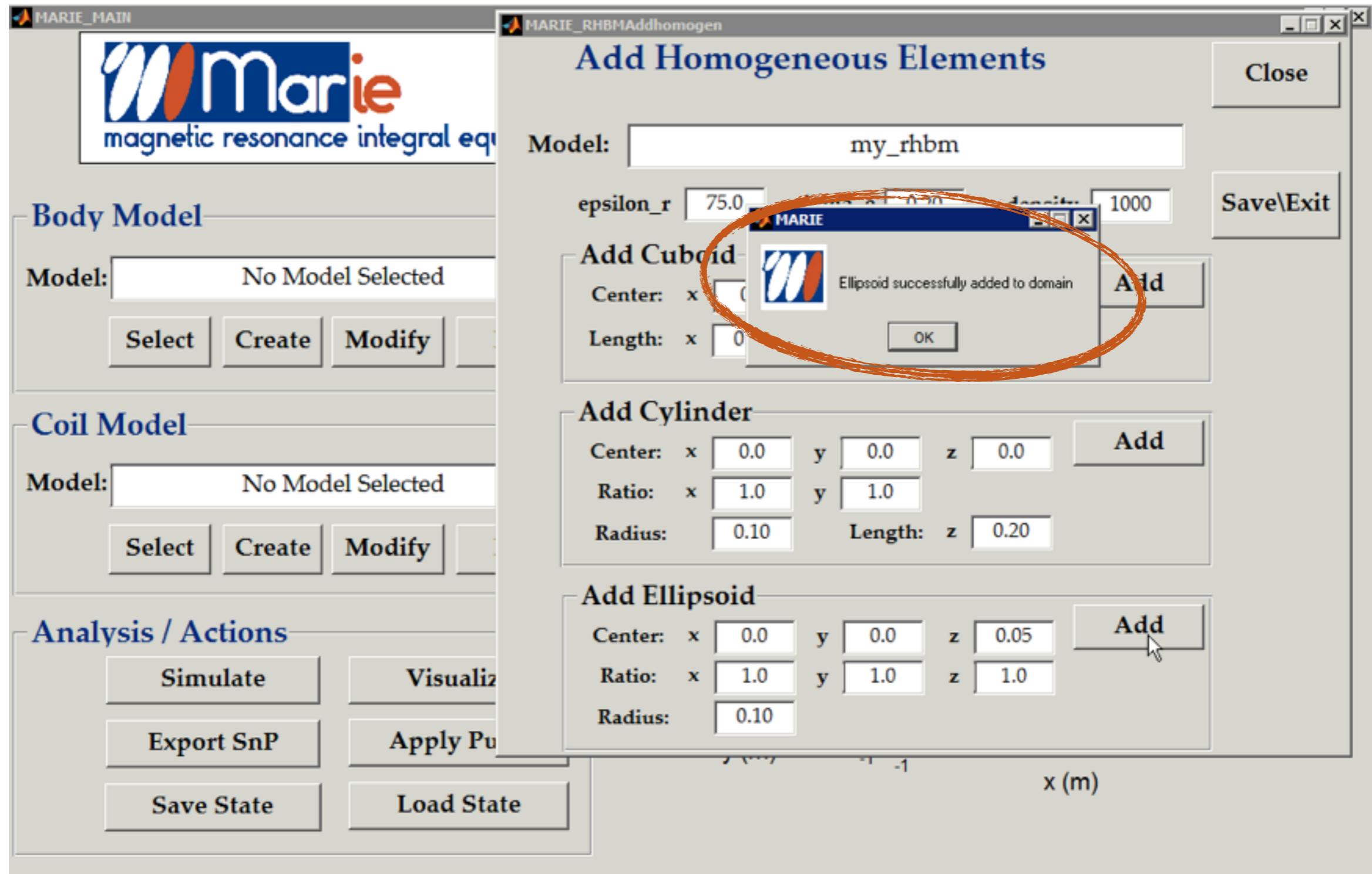


- ◆ Create a body model
  - ◆ same functionality exists for ellipsoids



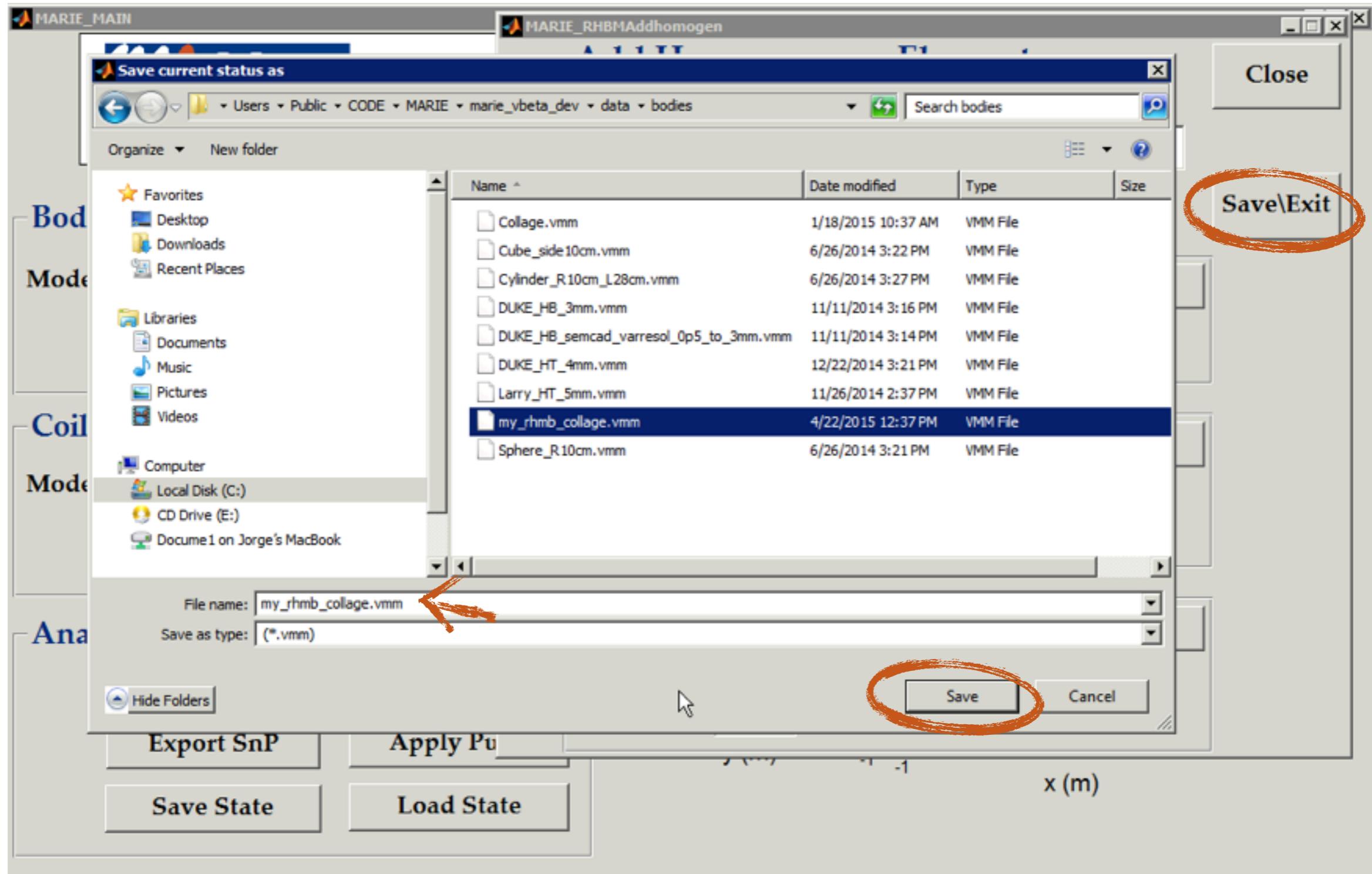
## ♦ Create a body model

- ♦ a window will indicate you have added the element



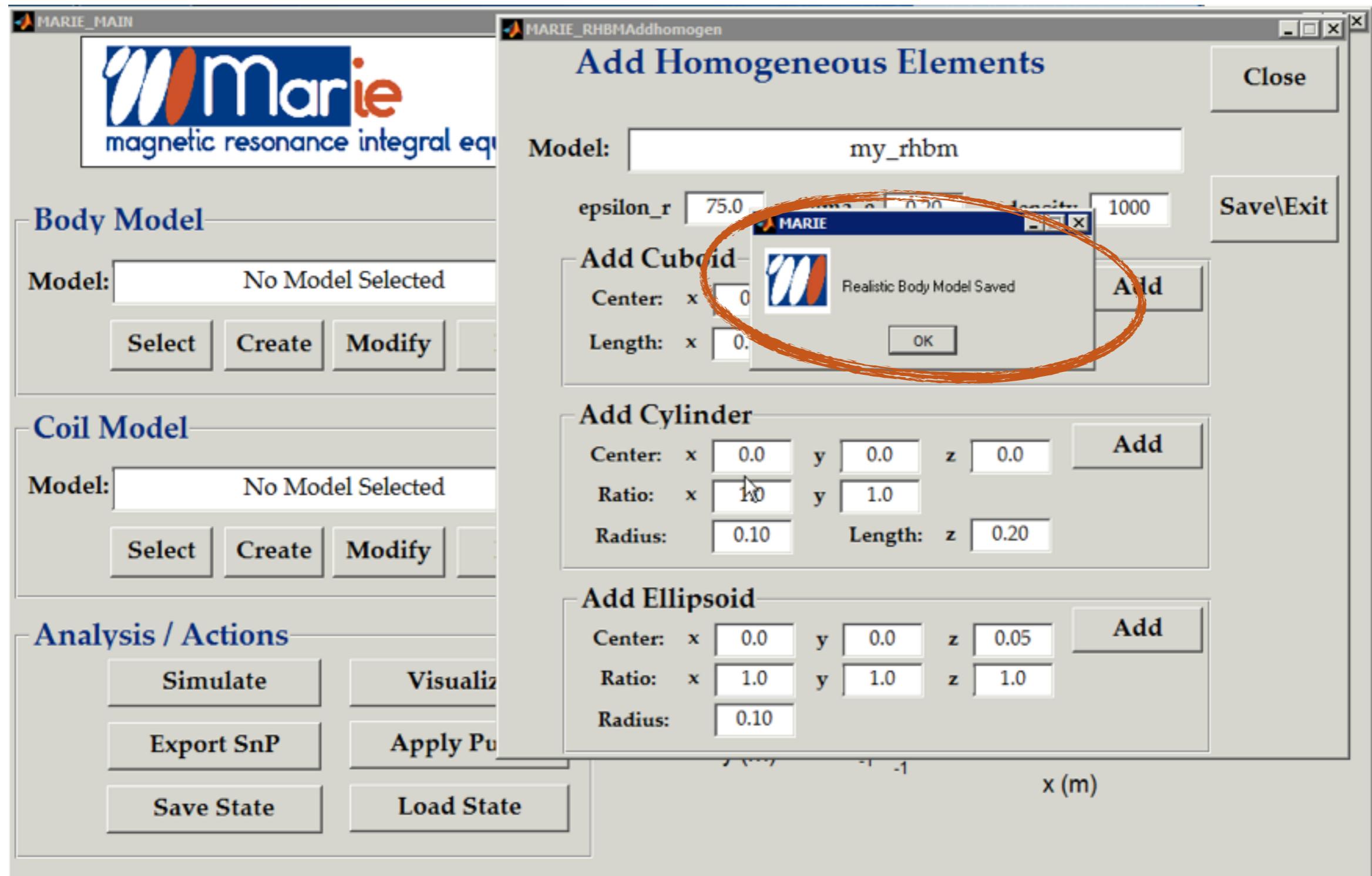
## ◆ Create a body model

- ◆ push “save\exit” to save the model in a .vmm file

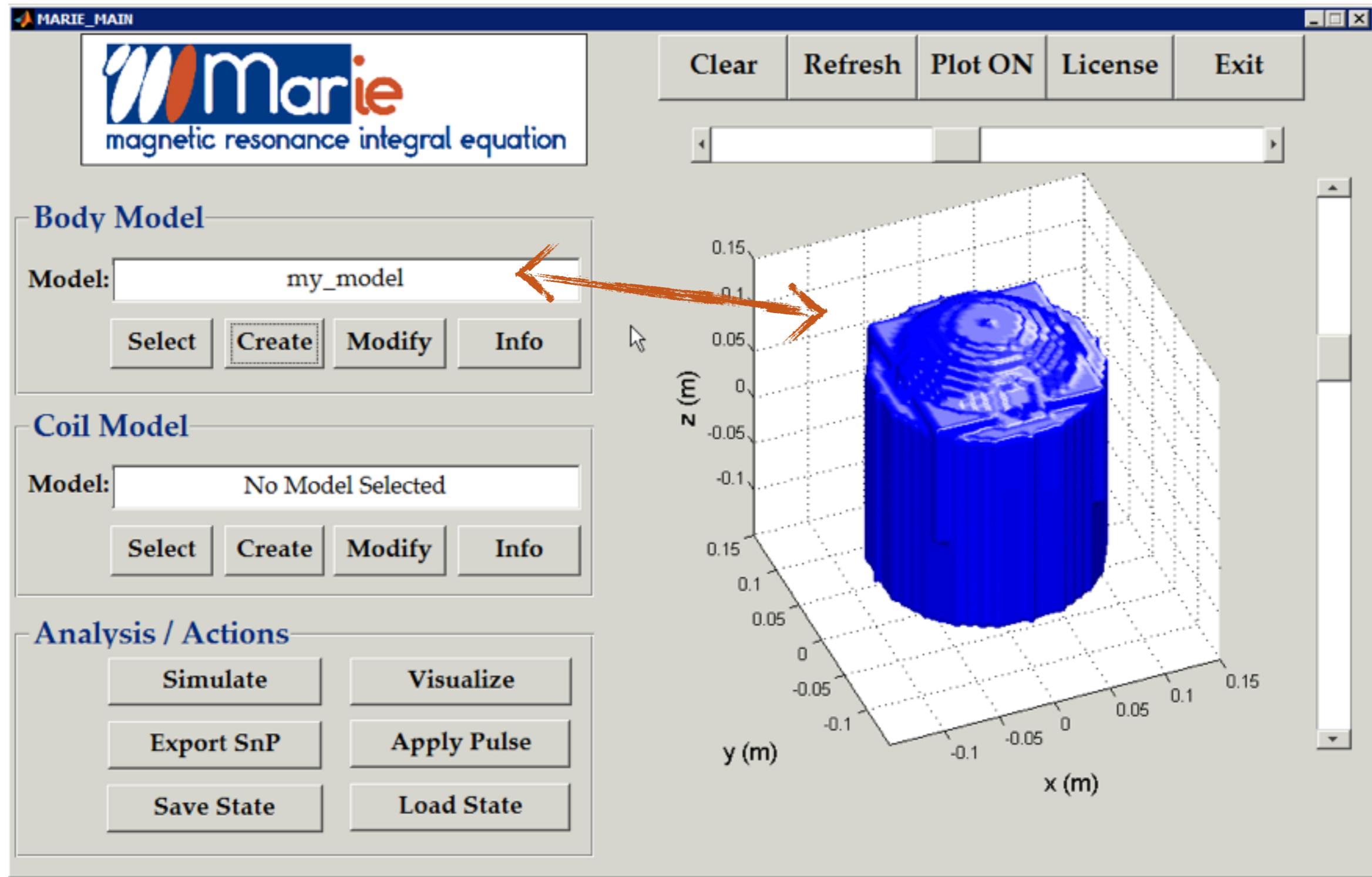


## ◆ Create a body model

- ◆ press save and a screen will indicate the model has been saved

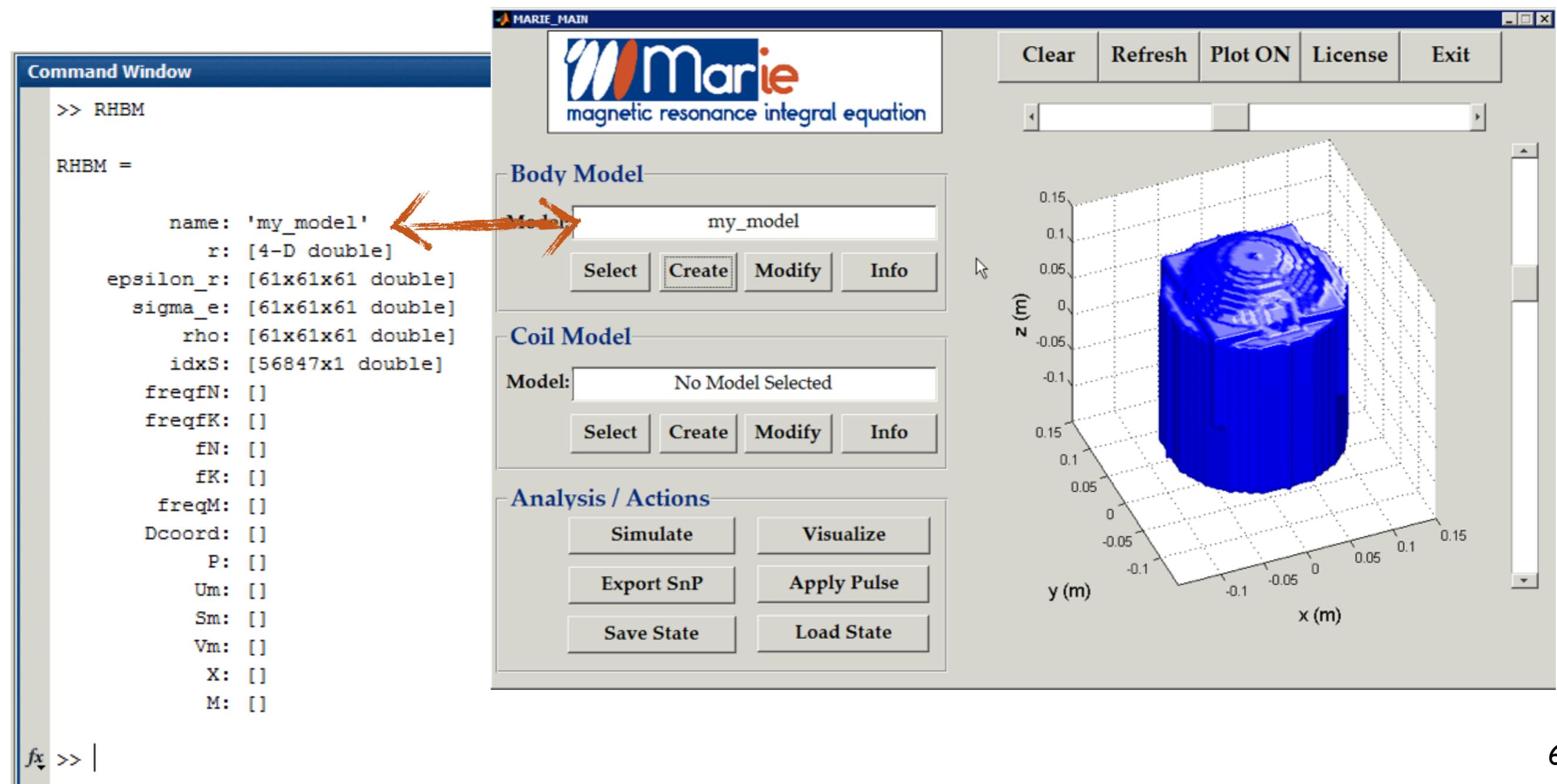


- ♦ Create a body model
  - ♦ the model will be loaded in the main window



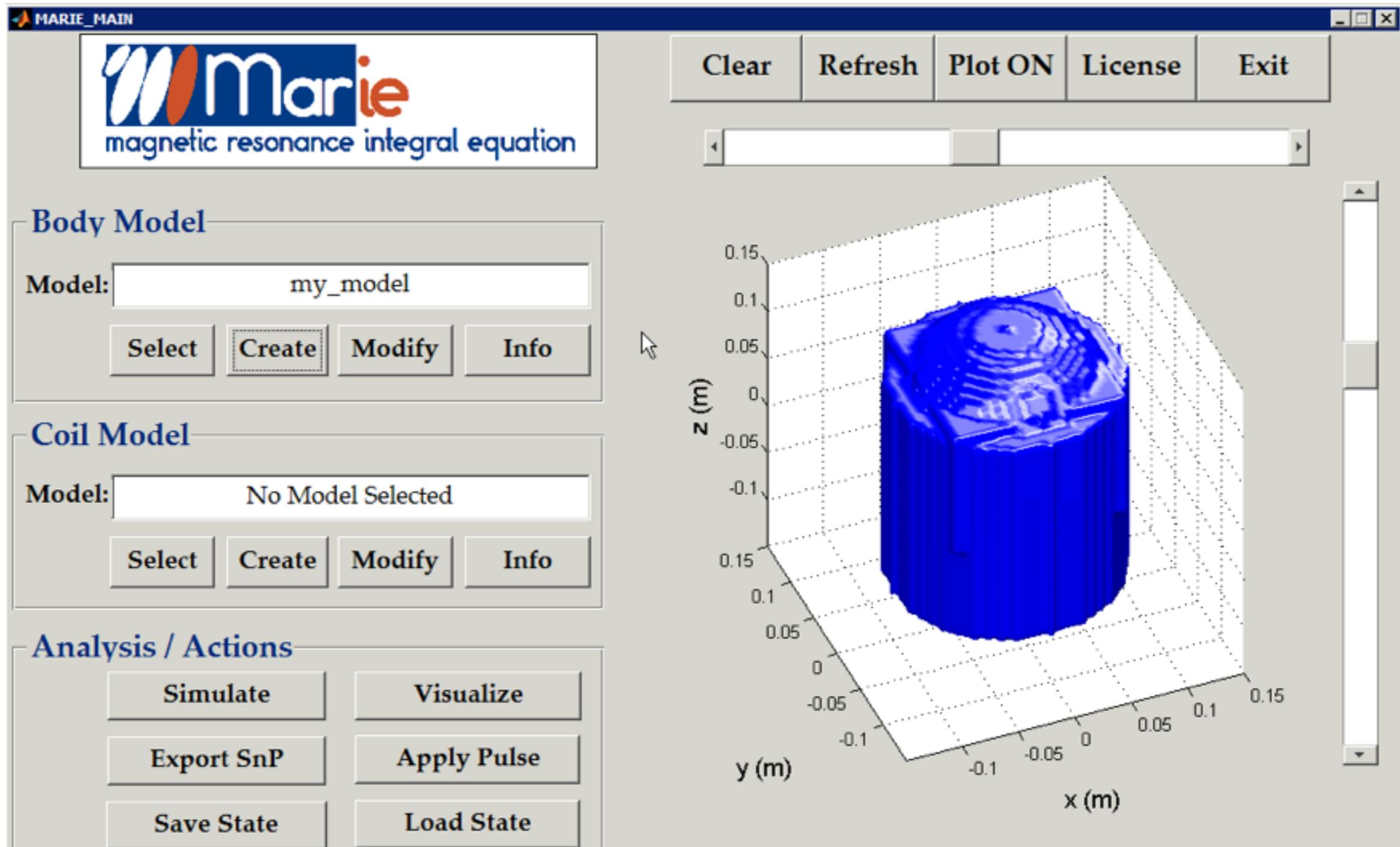
## ♦ Create a body model

- ♦ the model will be loaded in the main window
- ♦ you can see the updated RHBM structure in the command window

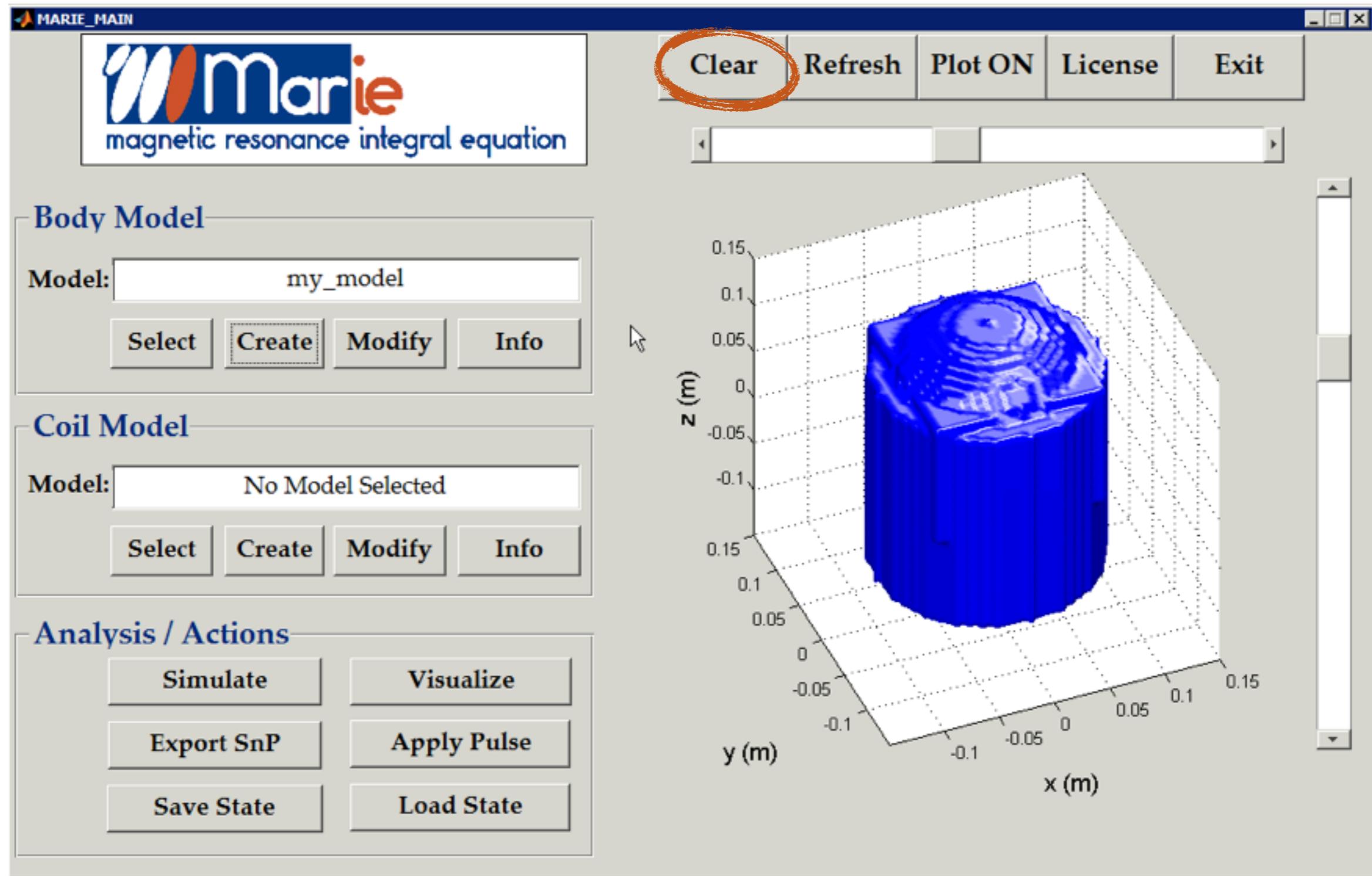


## ◆ Create a body model - important notes

- ◆ properties of latest shapes will overwrite existing voxel properties
- ◆ portions of shapes falling outside domain will be truncated

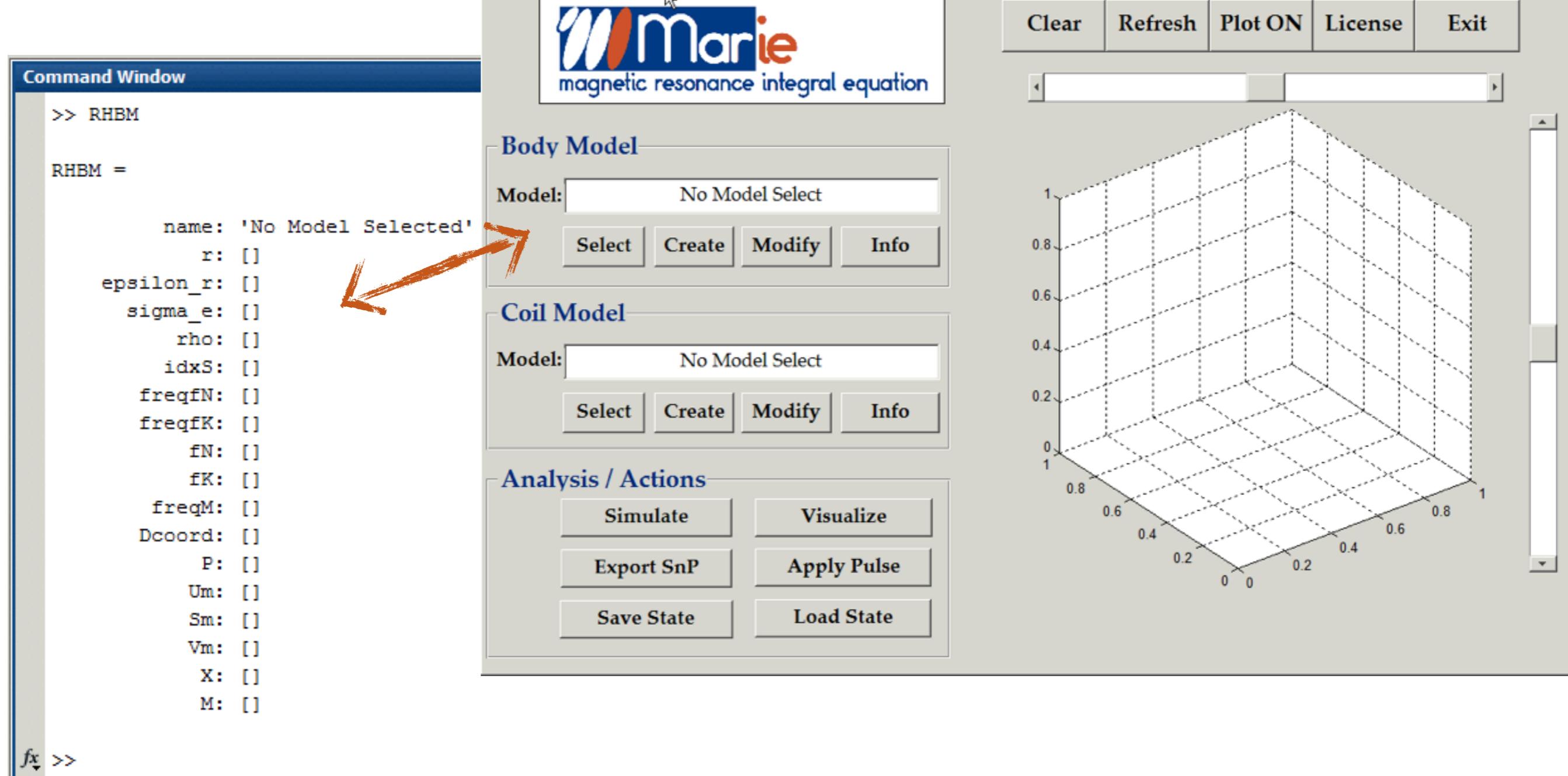


- ◆ Clear button
  - ◆ press the Clear button



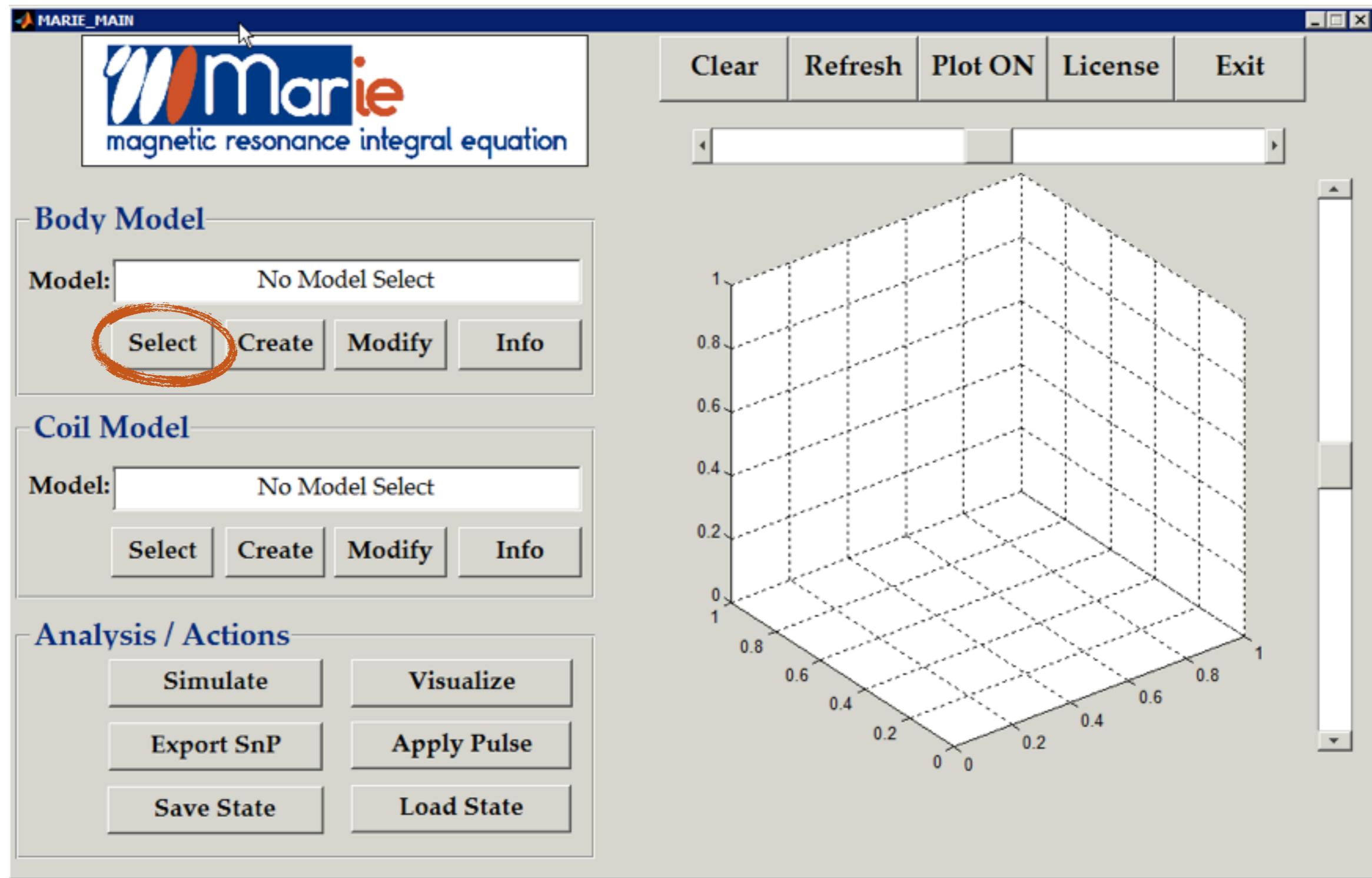
- ◆ Clear button

- ◆ it will clear all the structures of MARIE, removing models



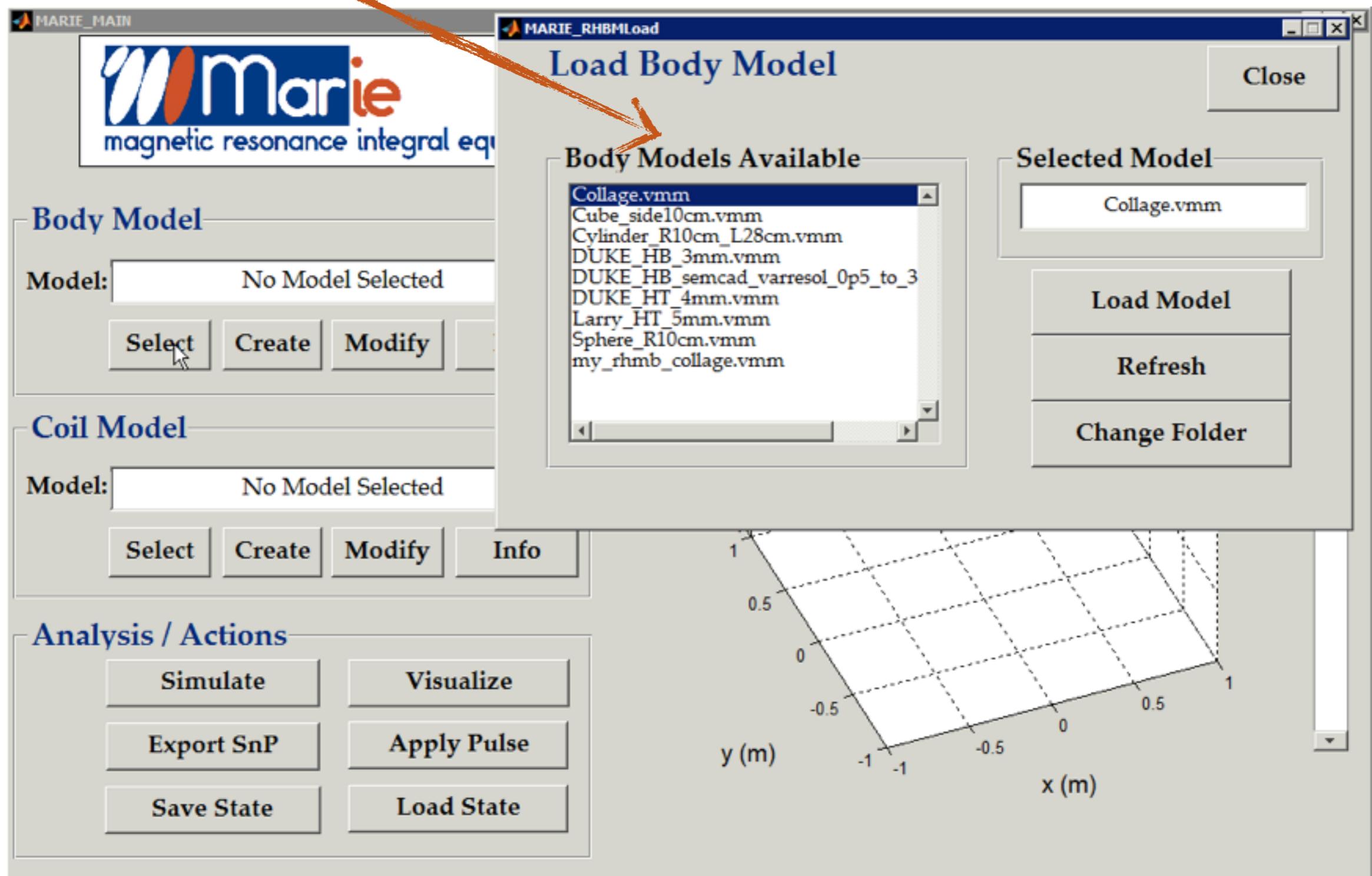
## ◆ Select body model

- ◆ You can load previously saved (or external) models in .vmm files



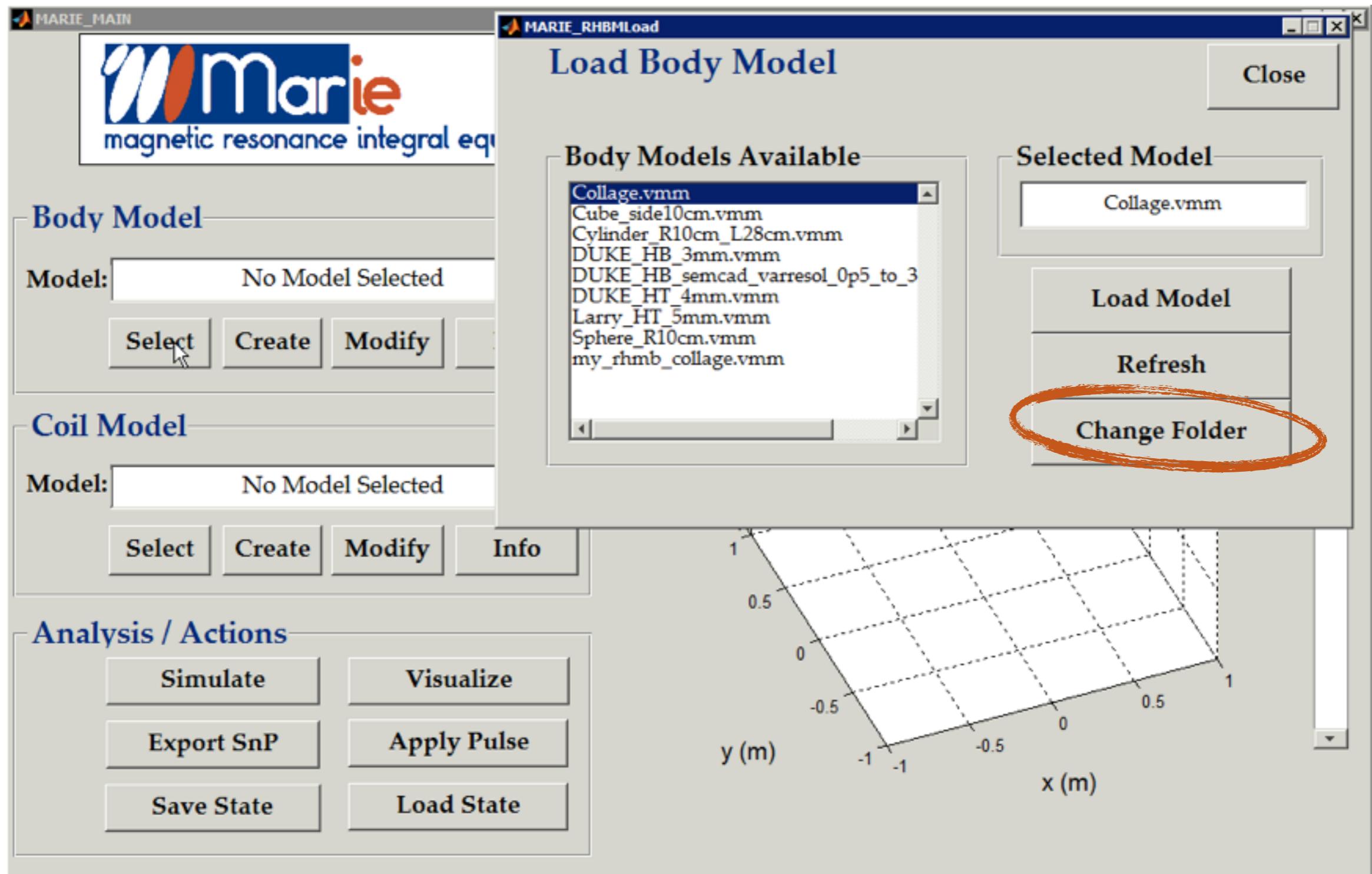
## ◆ Load body model

- ◆ available models in the default folder are shown



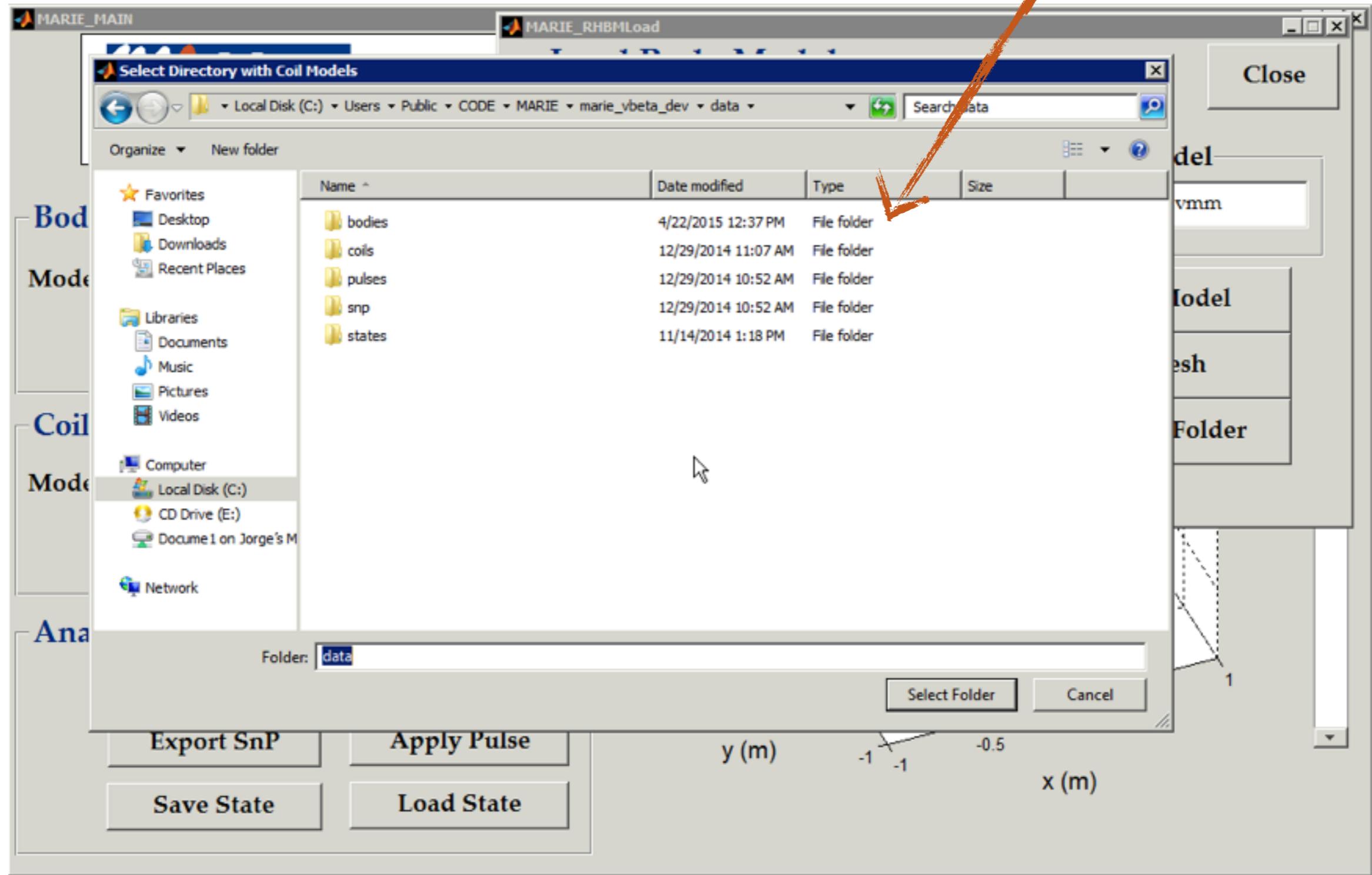
## ◆ Load body model

- ◆ push “change folder” to select a new folder



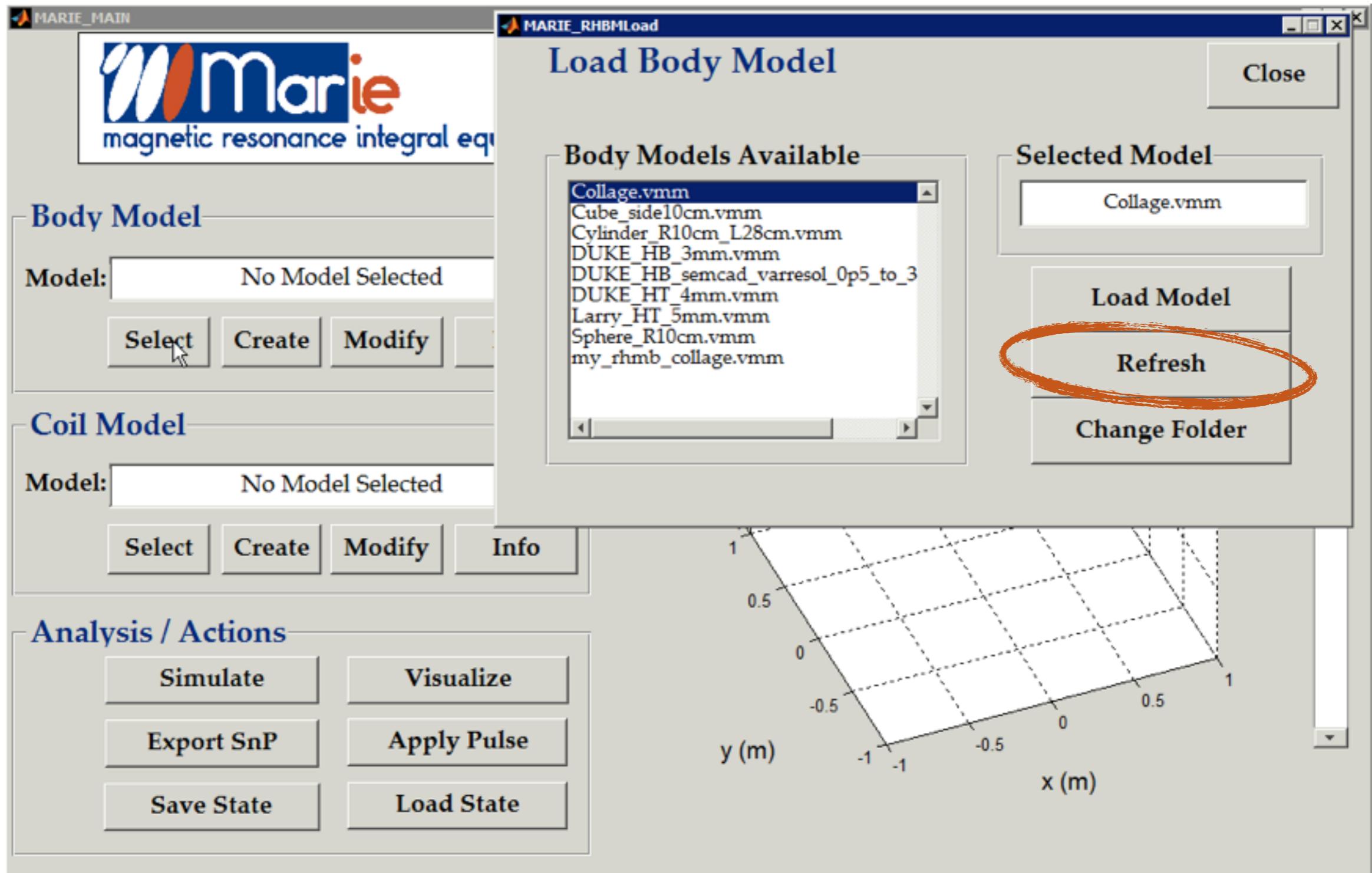
## ◆ Load body model

- ◆ a windows-like folder selection will appear: navigate to your folder



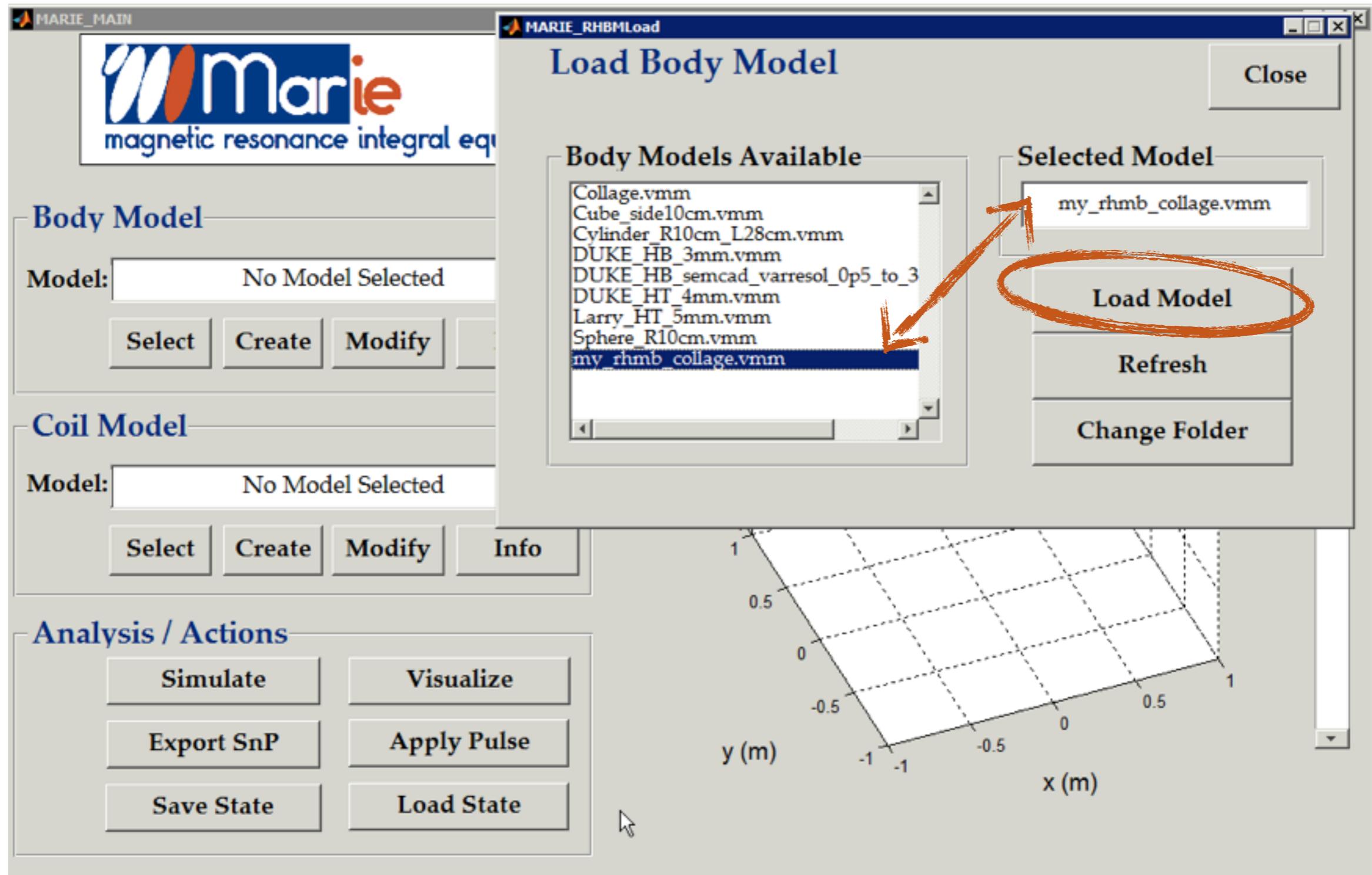
## ◆ Load body model

- ◆ “refresh” button updates the list (useful if new model file added)



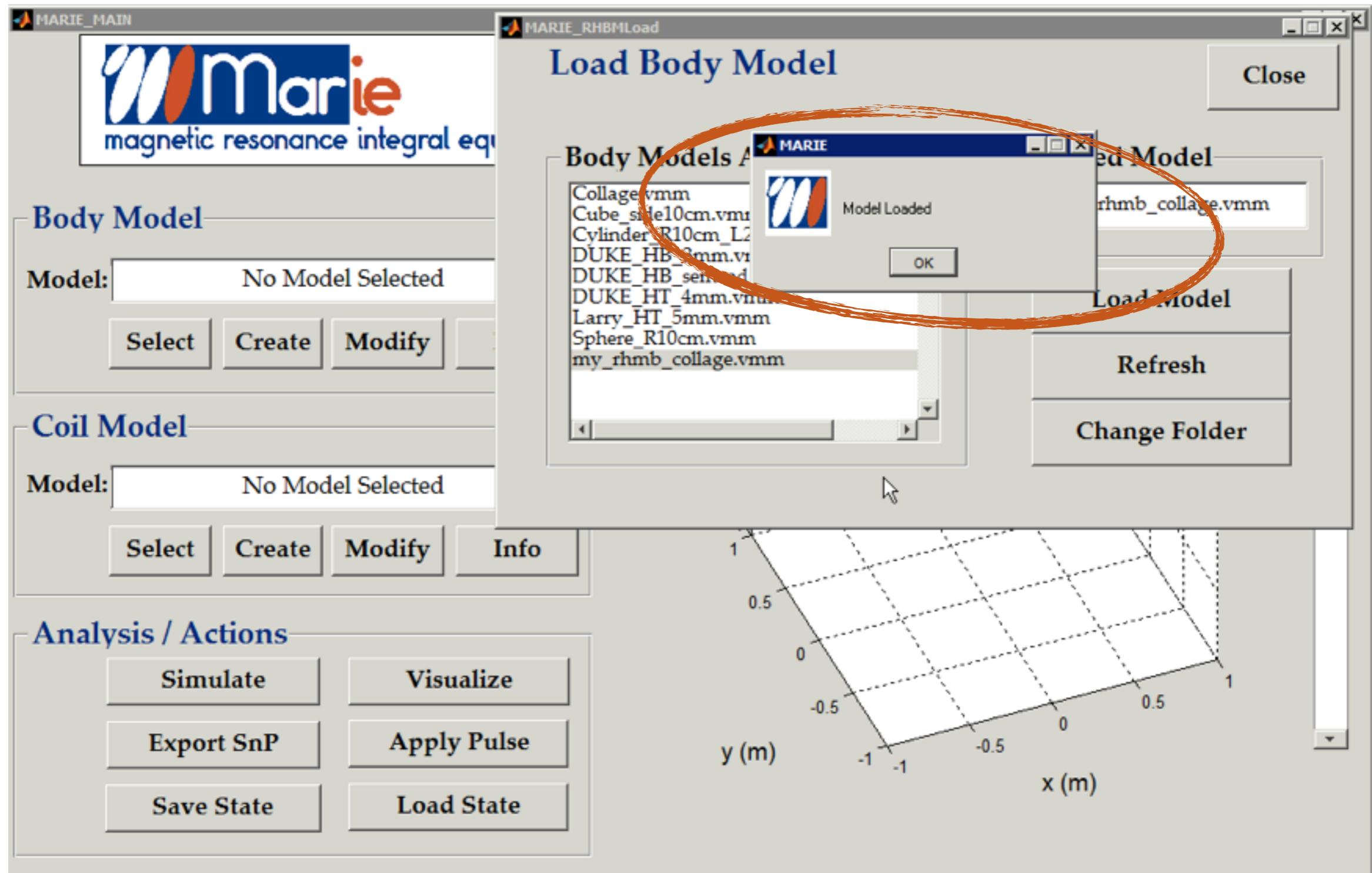
## ◆ Load body model

- ◆ select the desired model and push “load model”



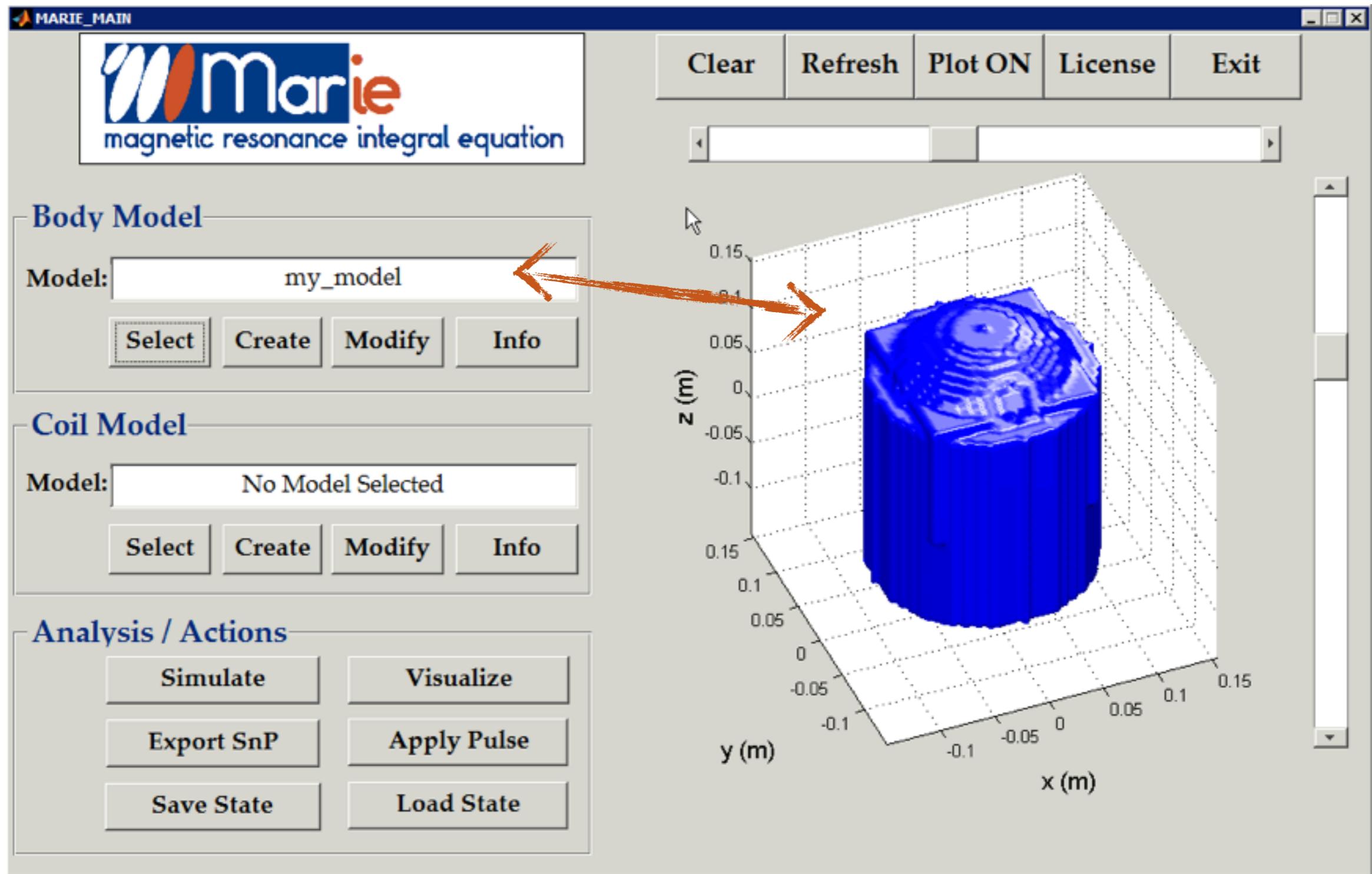
## ◆ Load body model

- ◆ a window will pop-up when the model is loaded



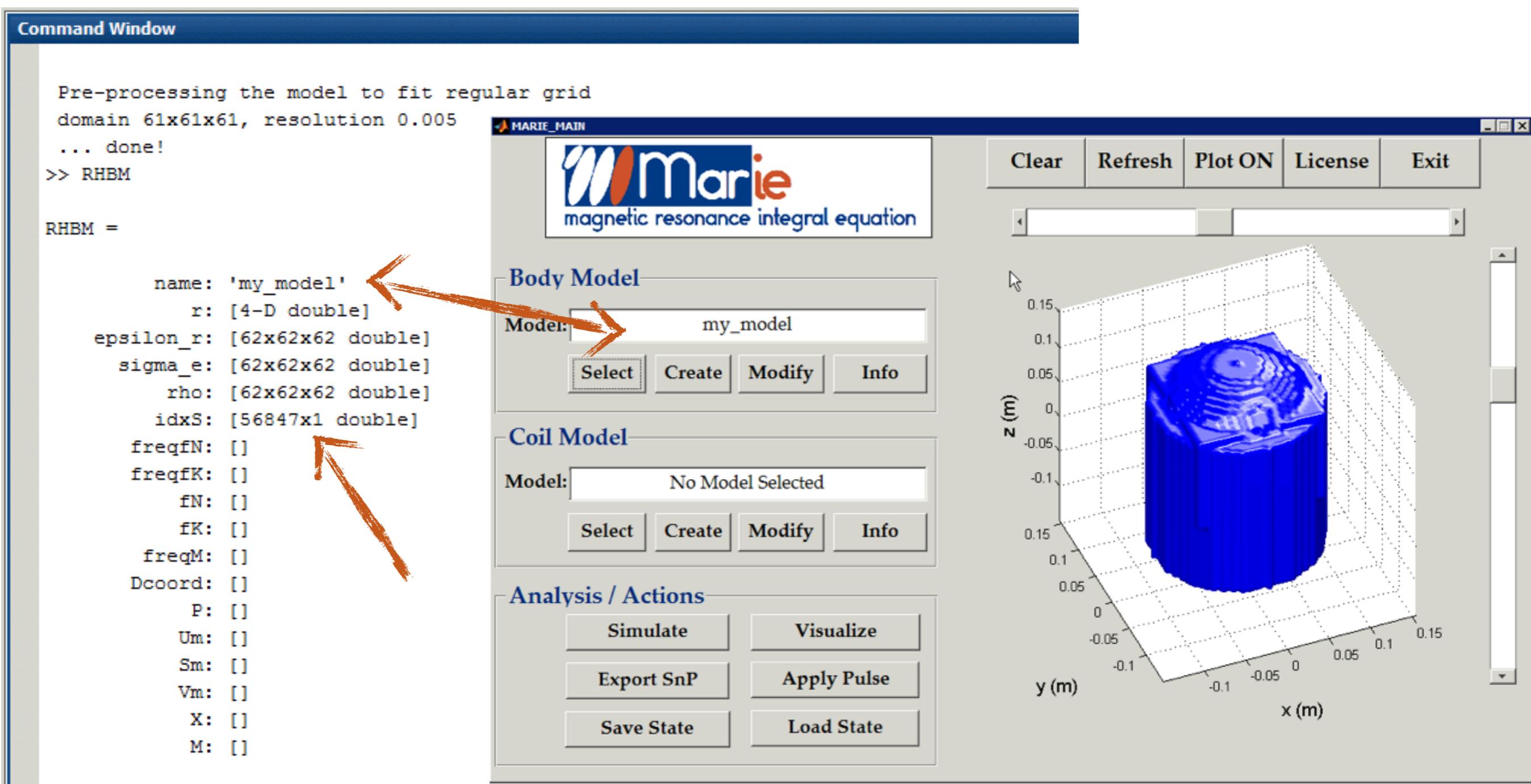
- ◆ Load body model

- ◆ and the model is loaded, shown in the figure



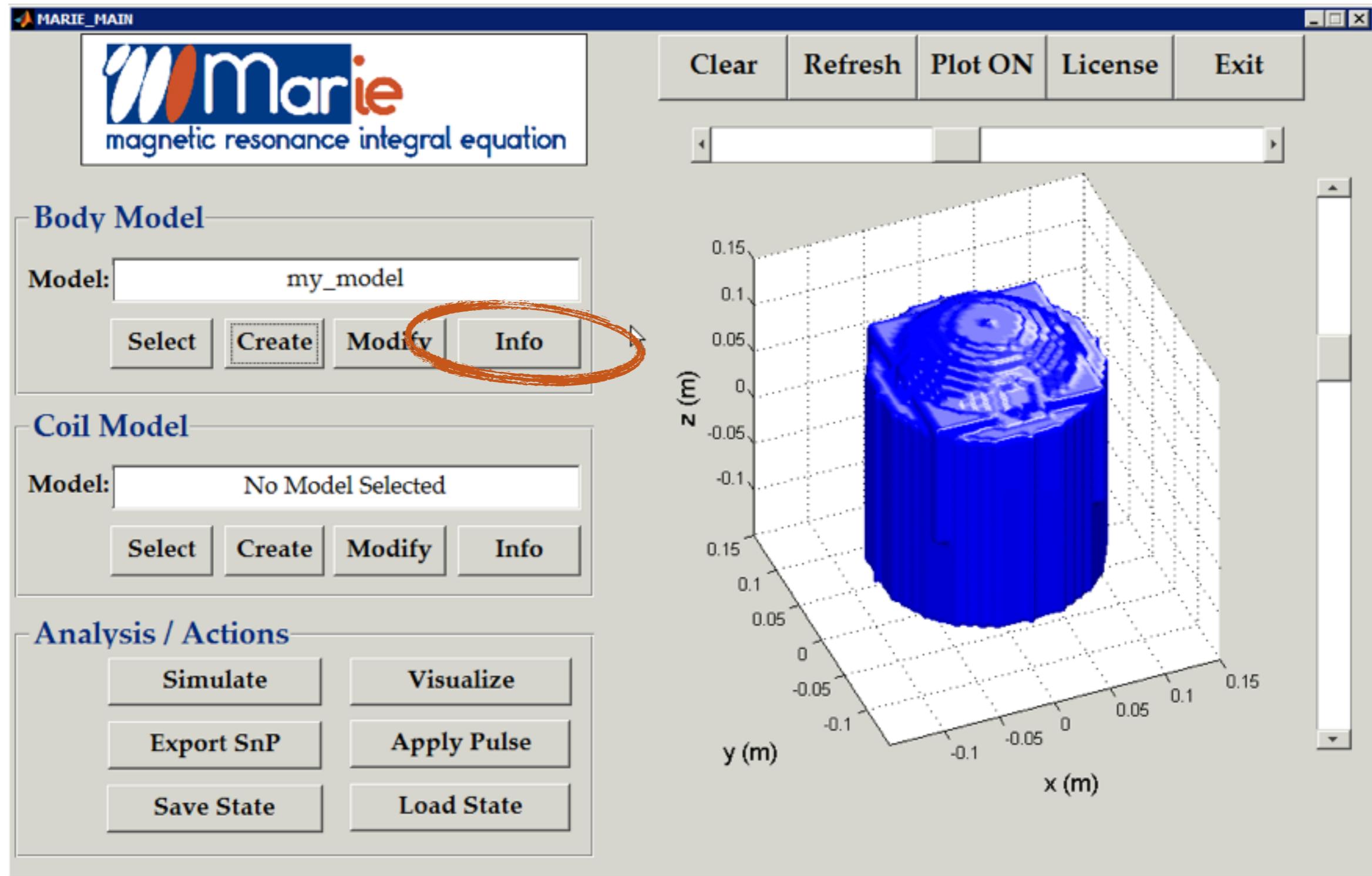
## ◆ Load body model

- ◆ data is stored in the RHBM structure

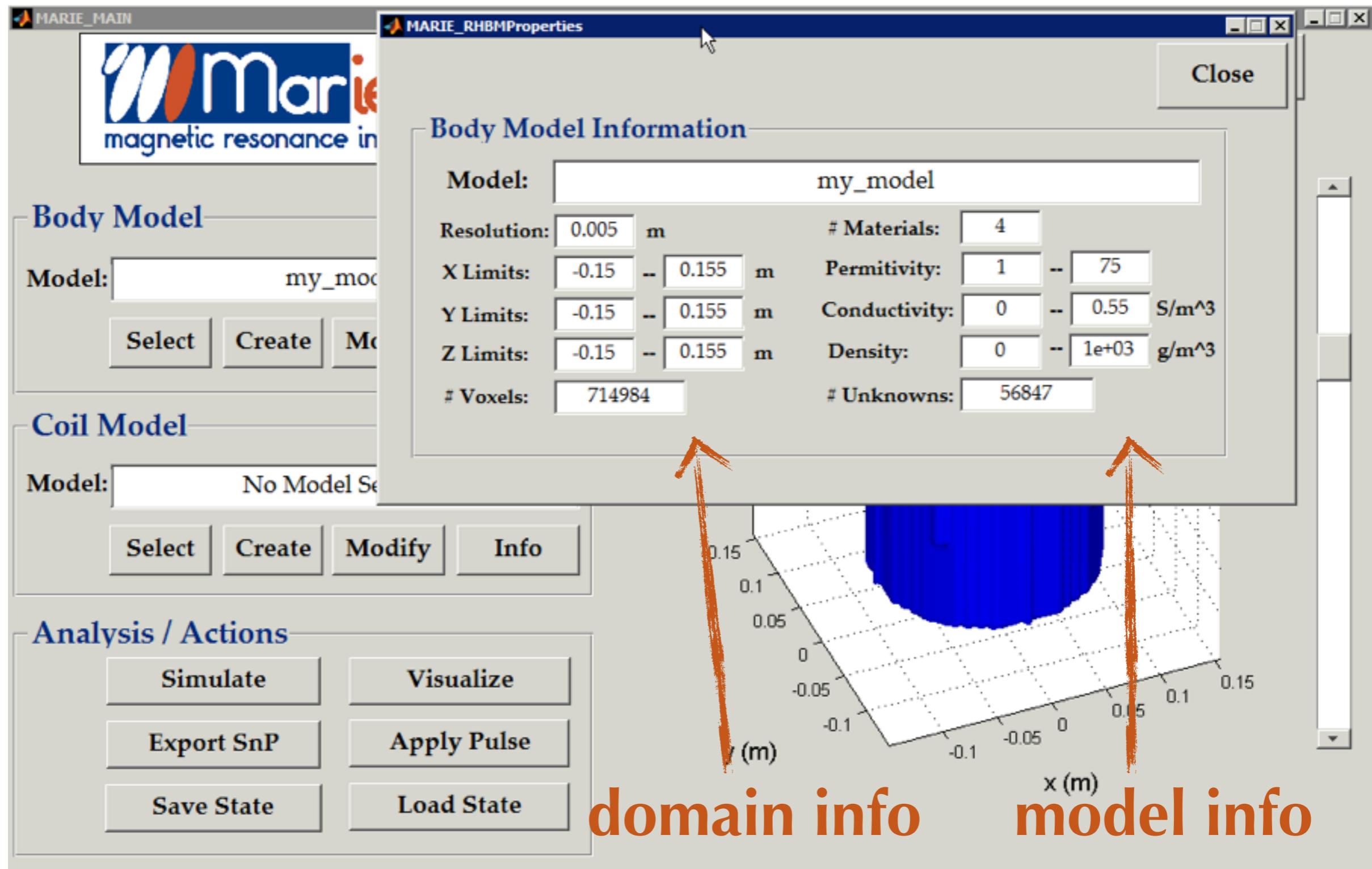


## ◆ Body model information

- ◆ push “info” button to get model information

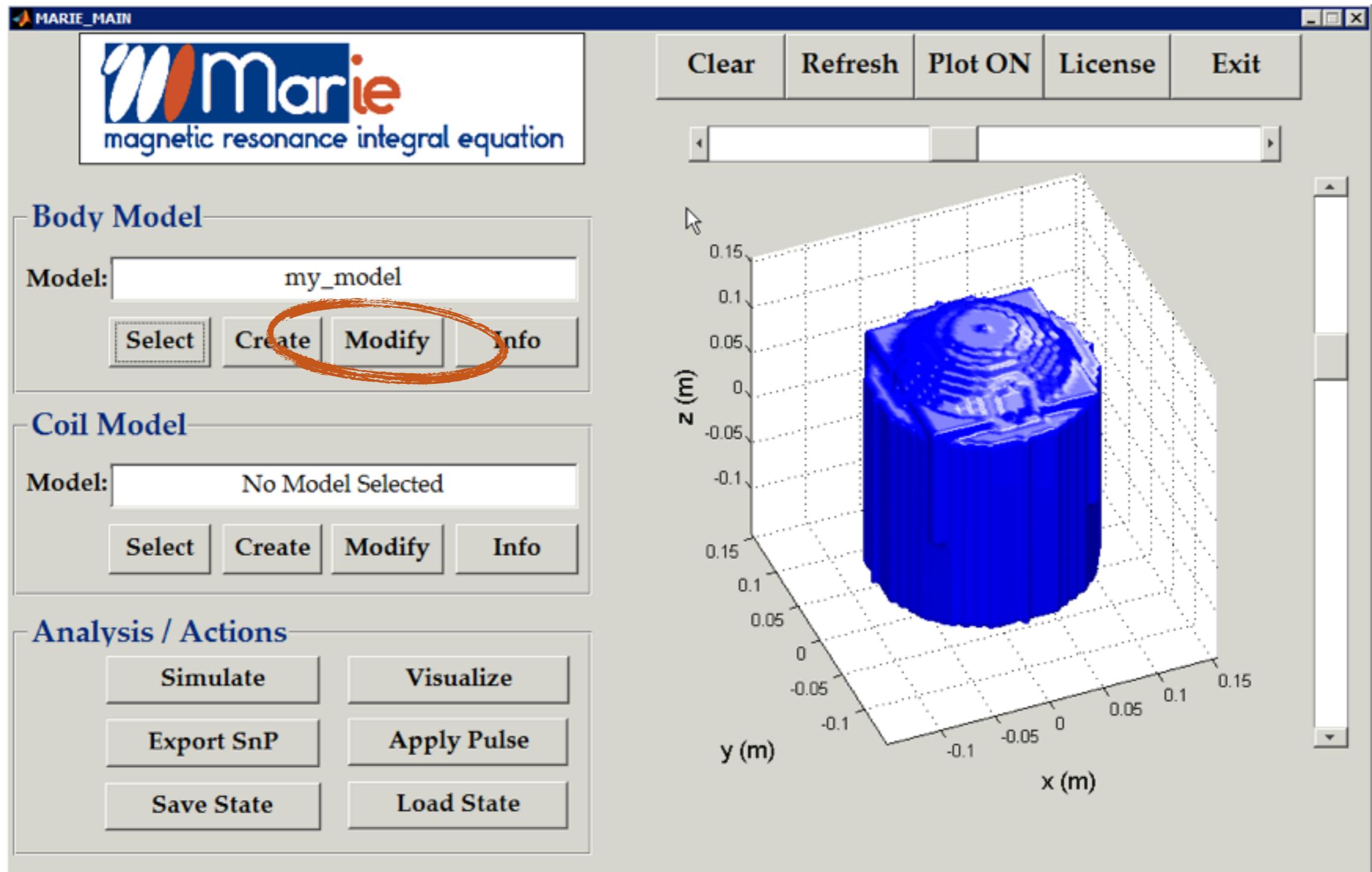


- ◆ Body model information
  - ◆ domain and model information will show



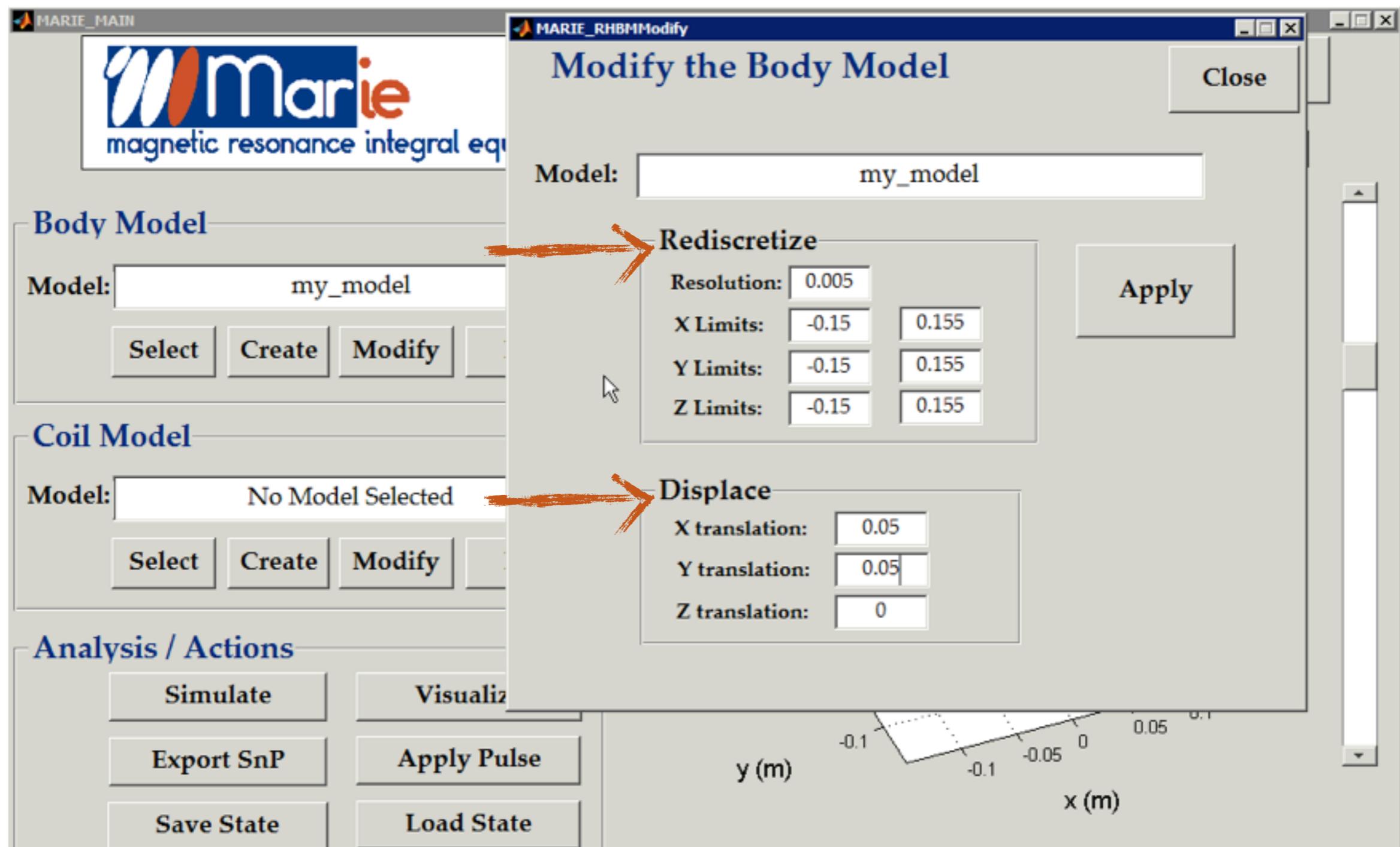
- ♦ **Modify the model**

- ♦ push “modify” to alter some characteristics of the model



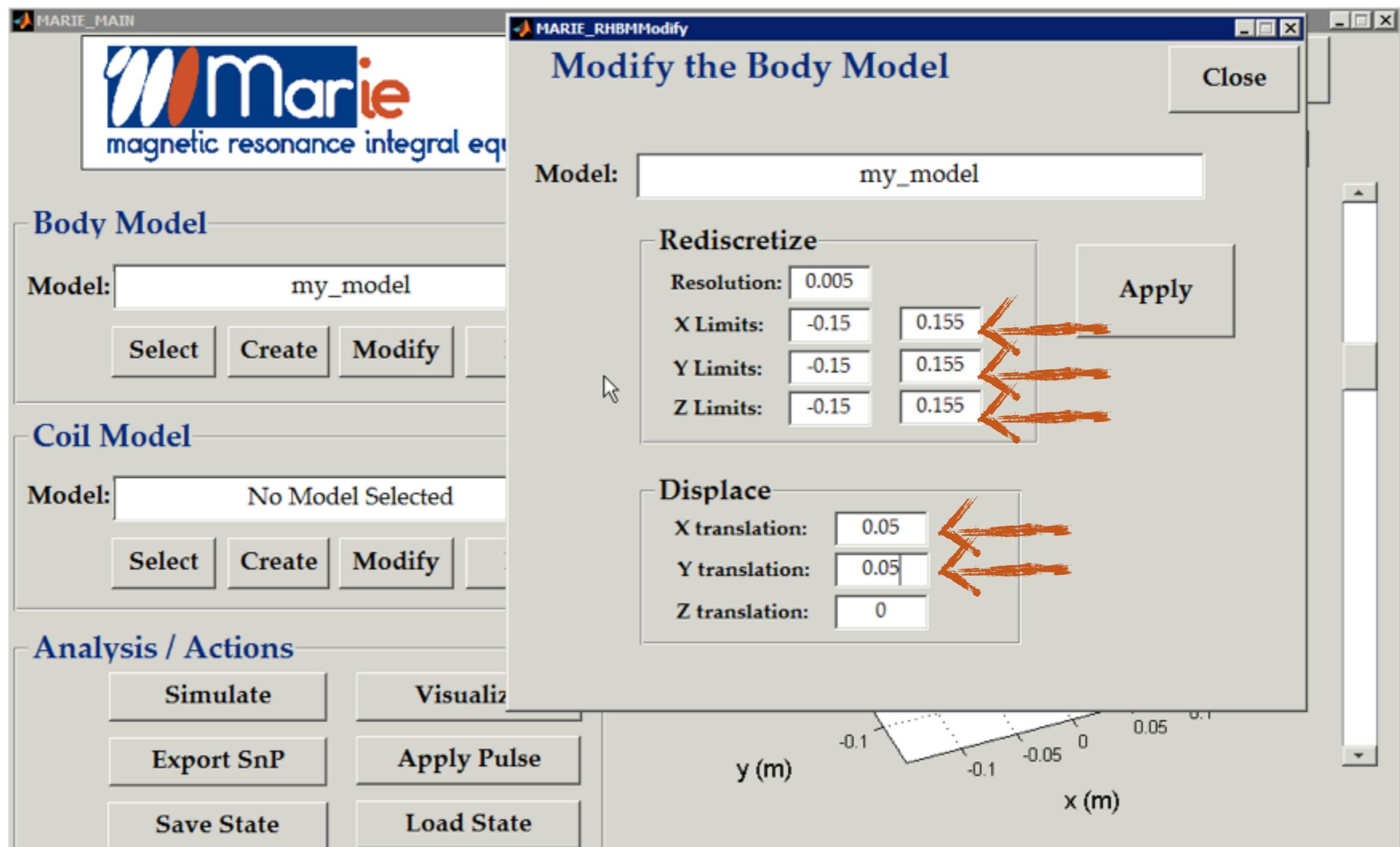
## ◆ Modify the model

- ◆ allows to modify the domain limits and change the resolution
- ◆ simple displacement of the coordinates of the domain

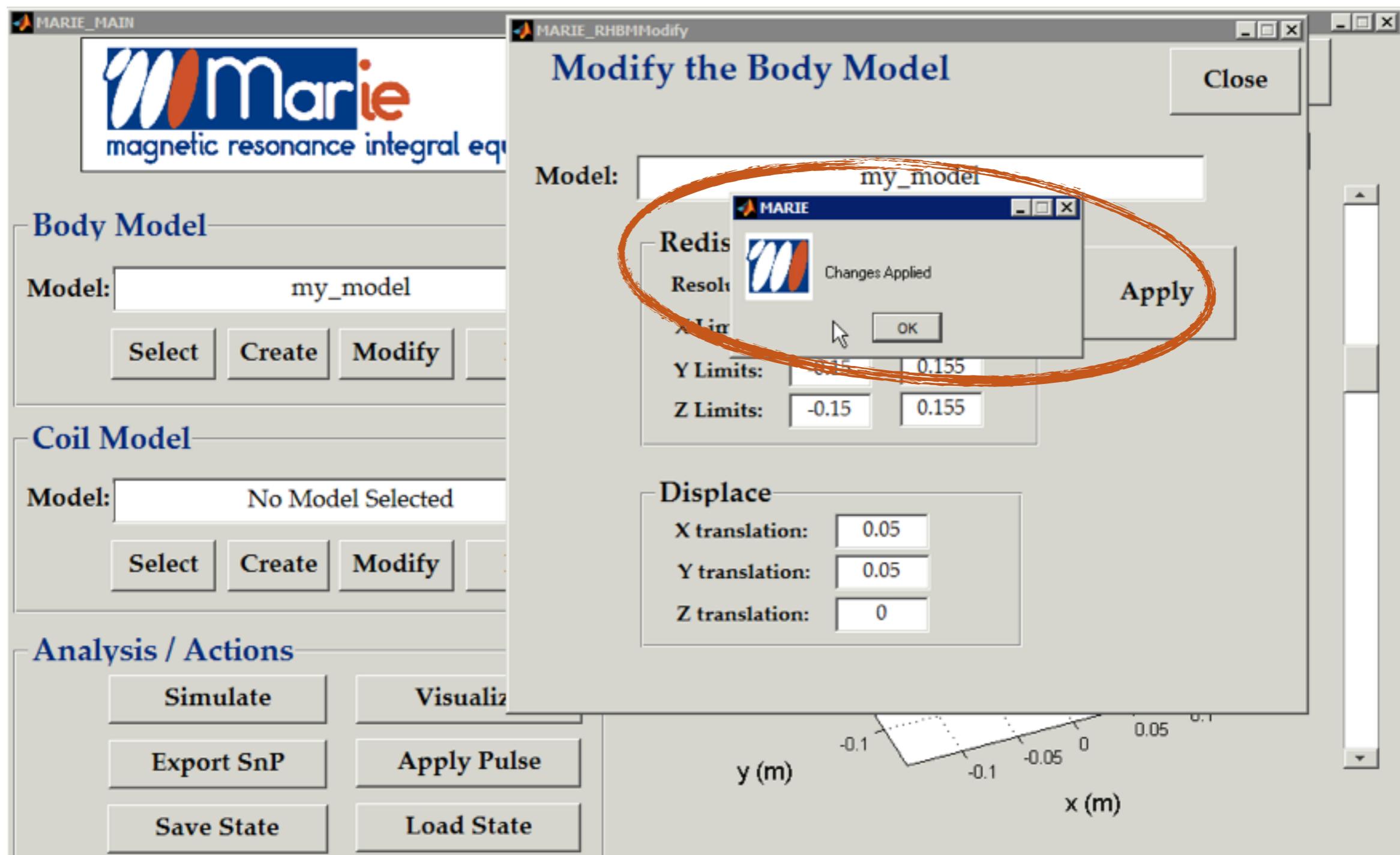


## ◆ Modify the model

- ◆ a simple extension of the domain, and displacement, push “apply”

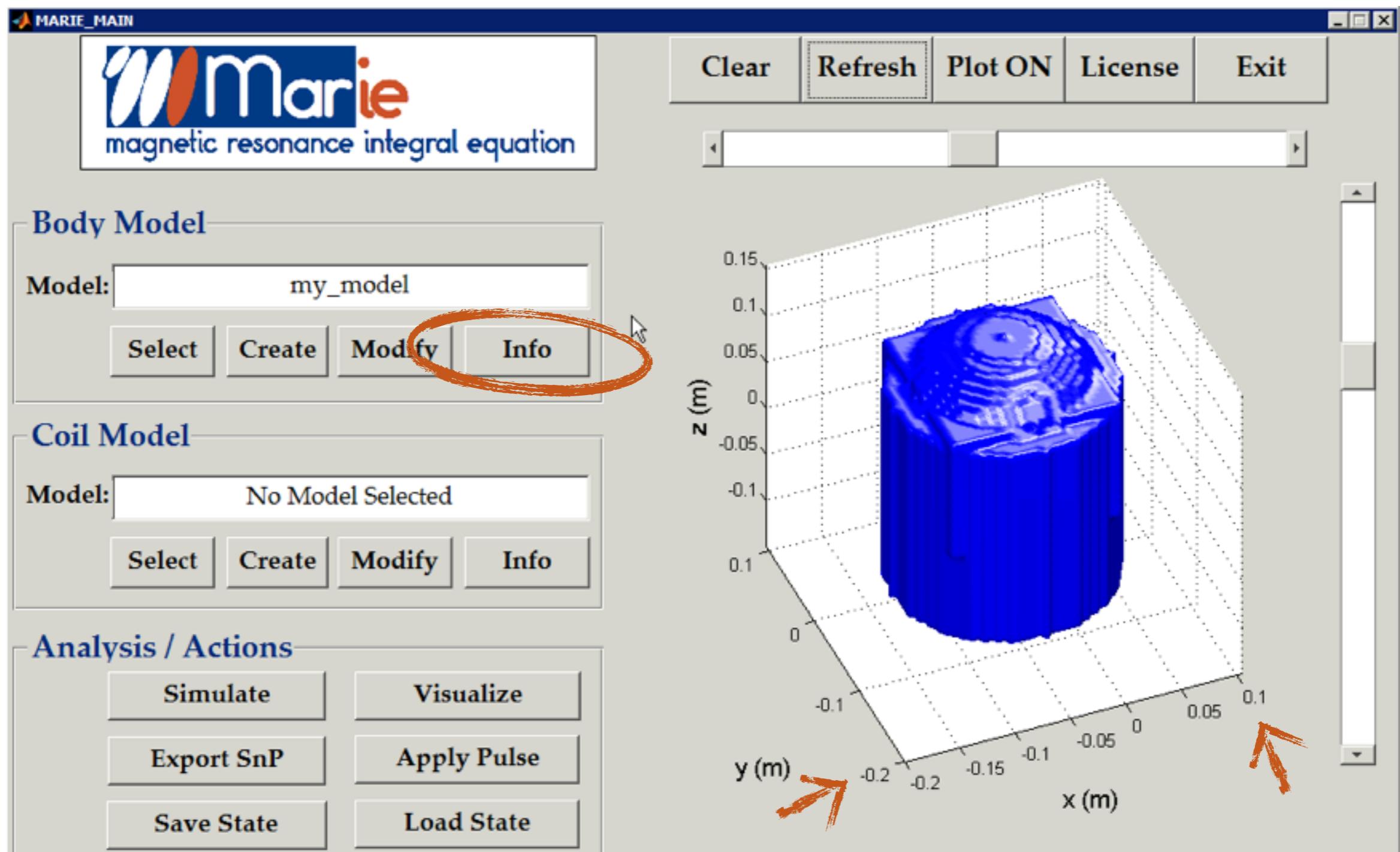


- ◆ **Modify the model**
  - ◆ the changes will be applied



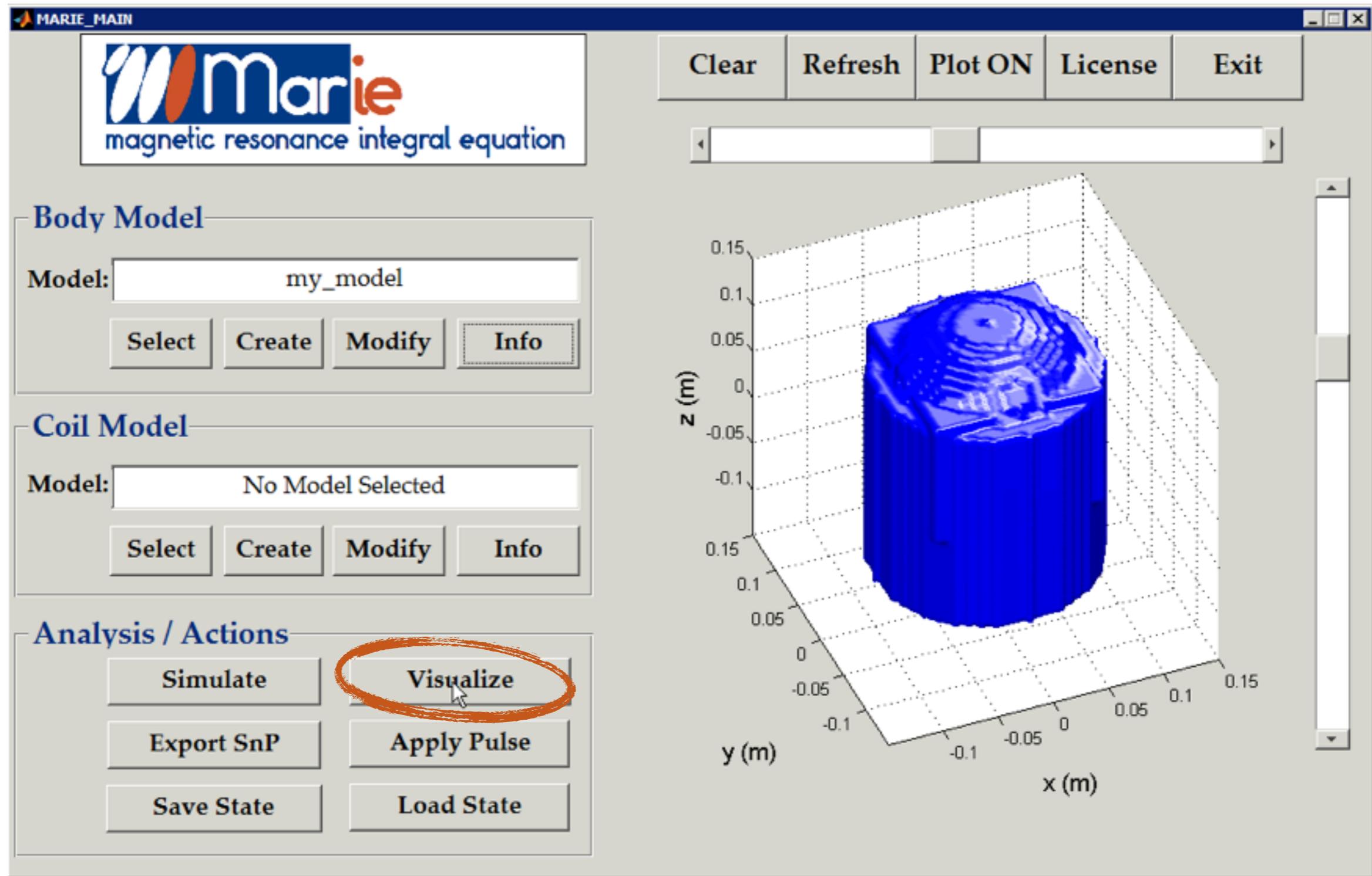
## ♦ Modify the model

- ♦ the changes can be seen in “info” (also note change in figure limits)

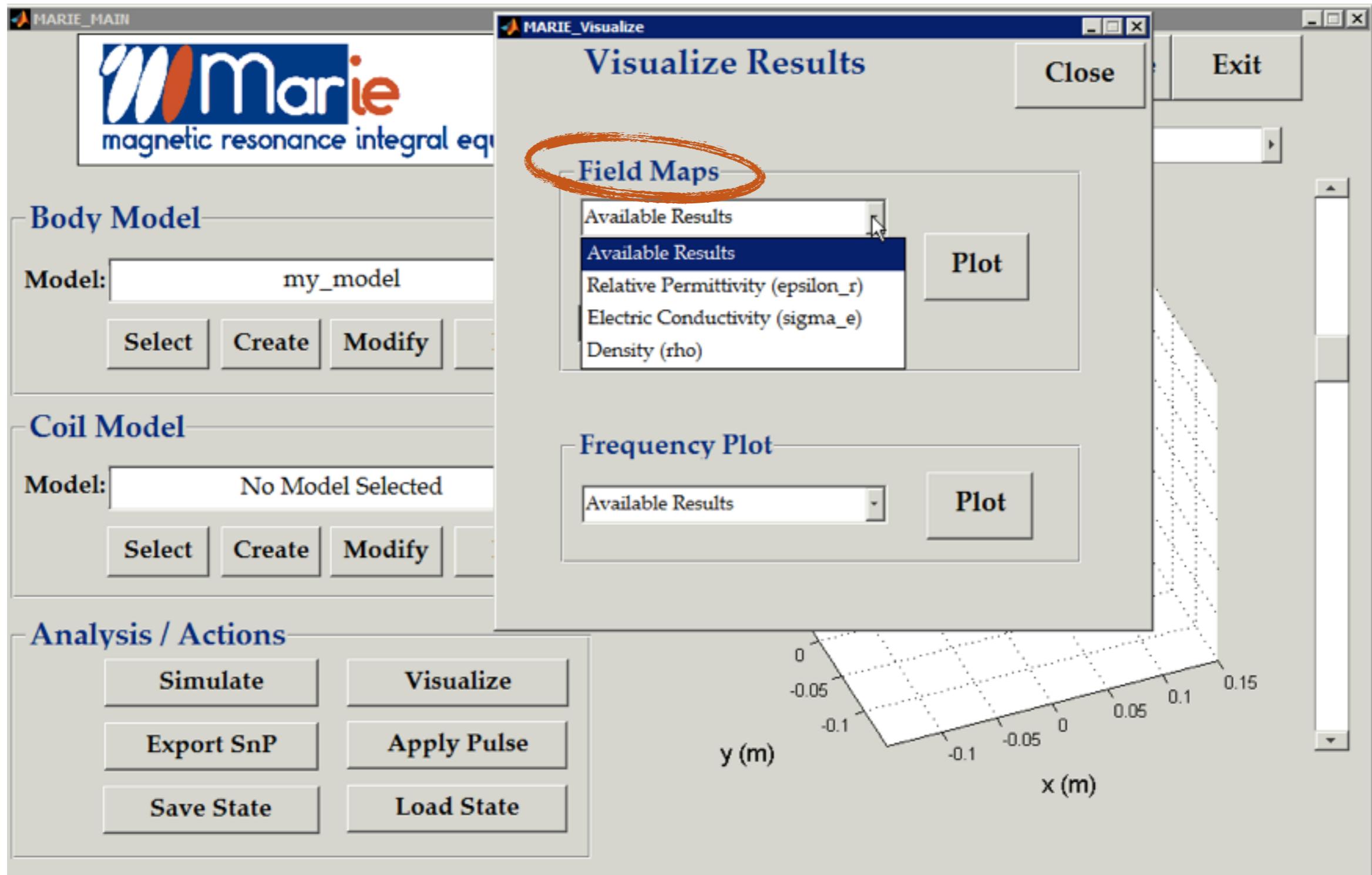


- ♦ Visualize the model

- ♦ press the visualize button to see the properties of the model

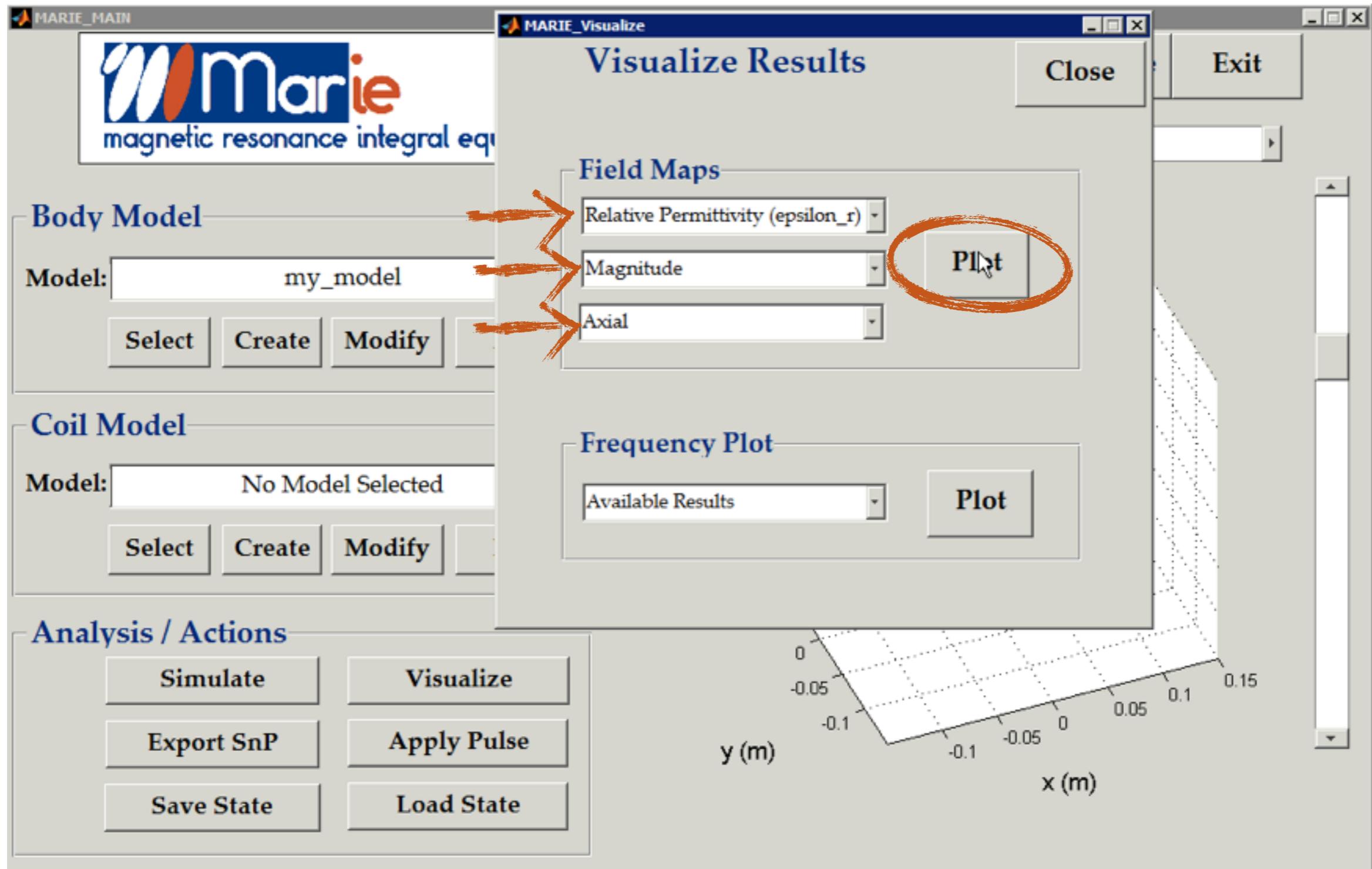


- ◆ Visualize the model
  - ◆ window for visualization will pop-up

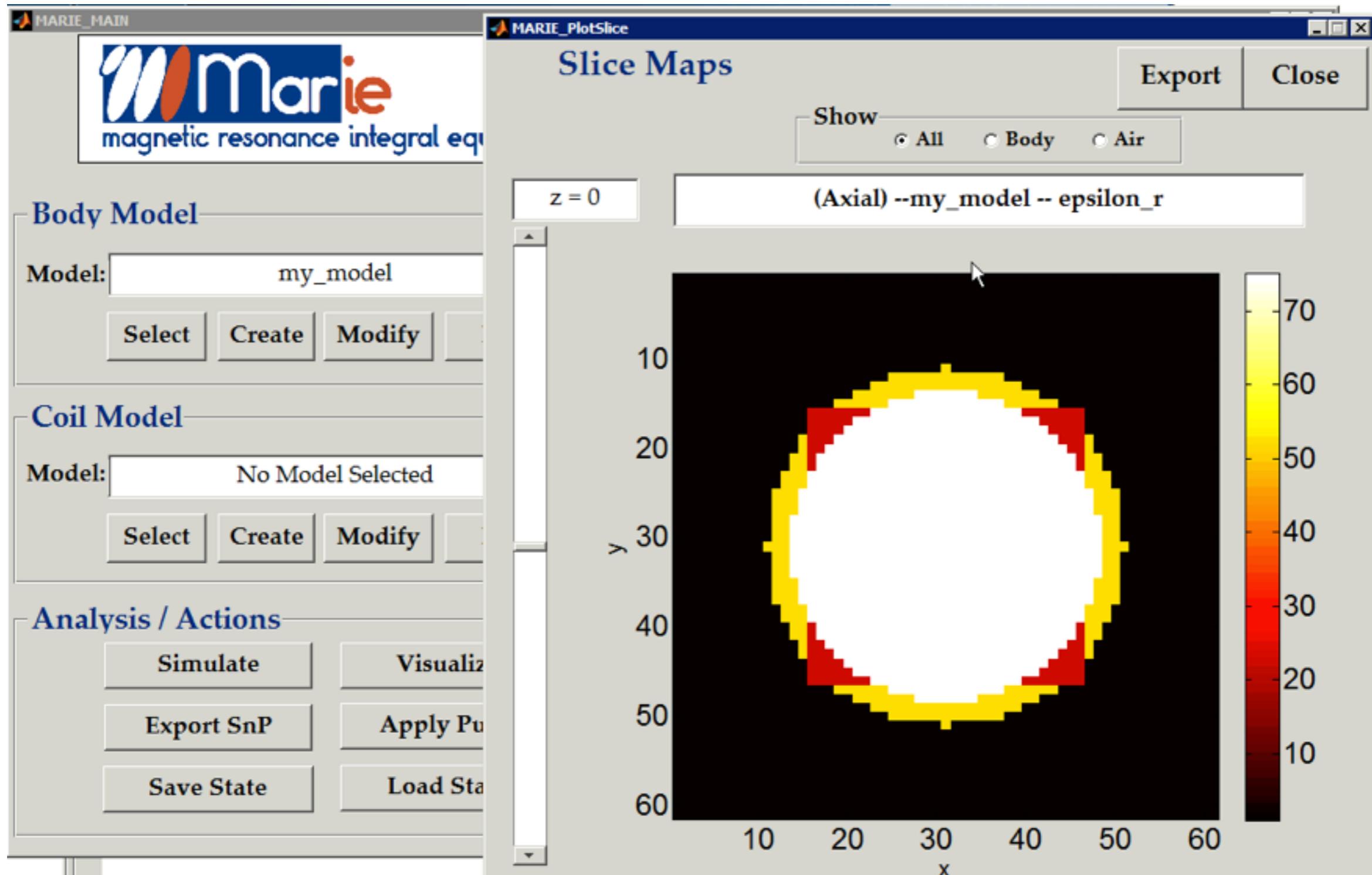


- ♦ Visualize the model

- ♦ select the map you want to plot

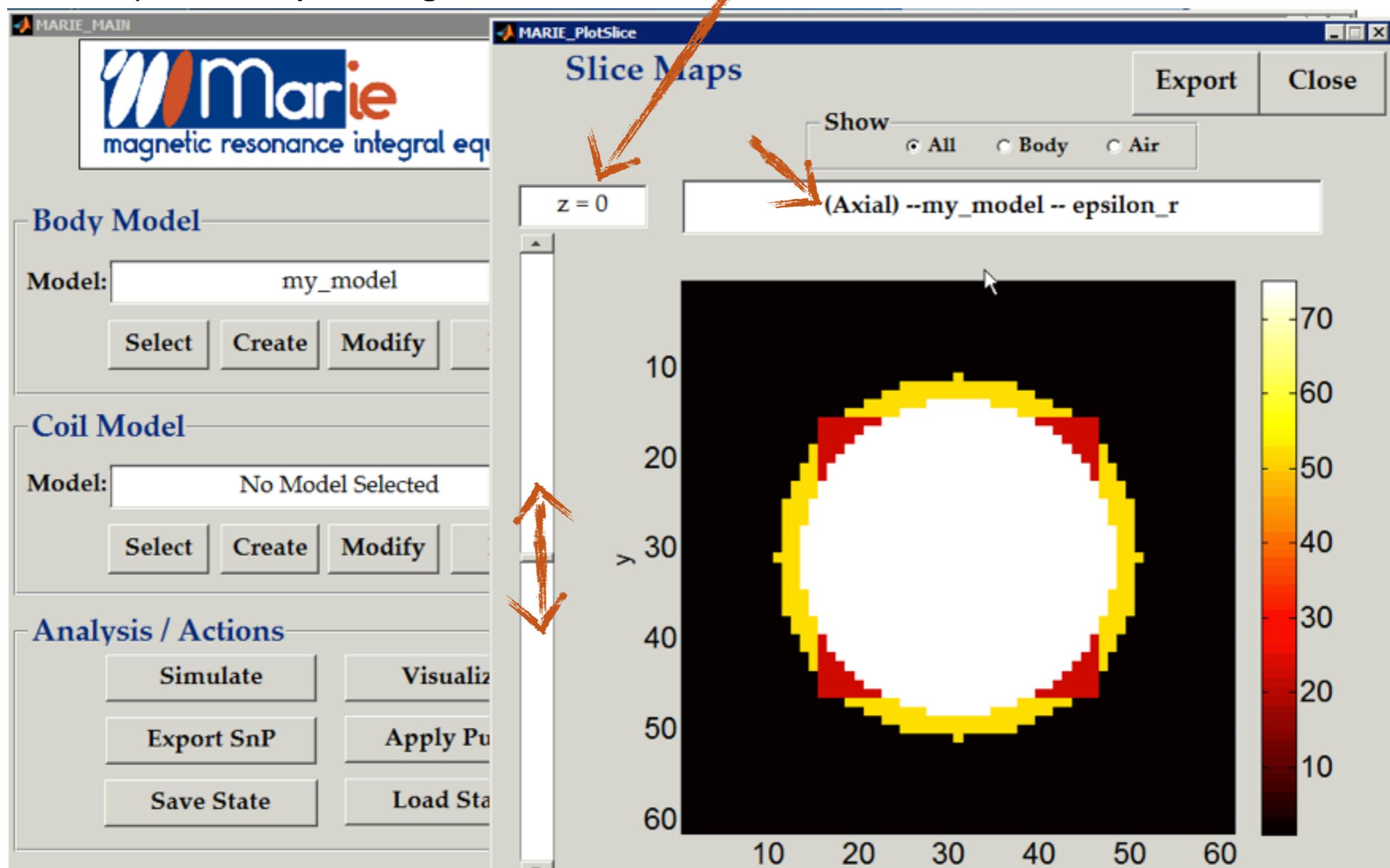


- ◆ Visualize the model
  - ◆ it will plot the corresponding map



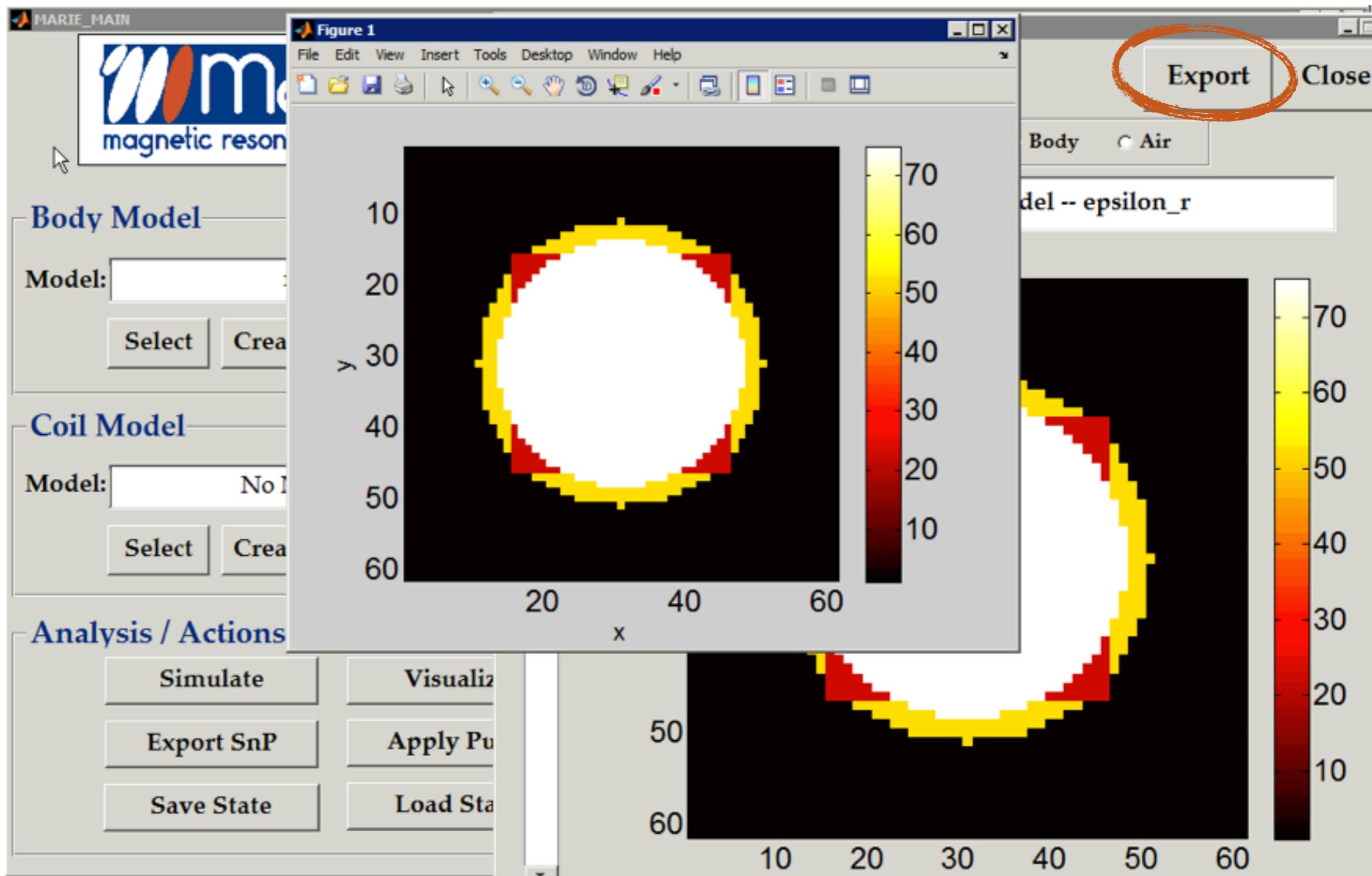
## ♦ Visualize the model

- ♦ you can scroll to different cut positions
  - ♦ x,y, or z depending on the view



- ♦ Visualize the model

- ♦ press “export” to generate a matlab figure, to be modified, saved...



gui - coil model box

## ◆ Function box related to the coil model

- ◆ the initial model is empty
- ◆ “No Model Selected” in the coil model field, figure empty

**Command Window**

```

>> whos
  Name      Size            Bytes  Class
  COIL      1x1              386   struct
  FIGIDX    1x1               8    double
  PULSE     1x1             1760   struct
  RHBM      1x1             1090   struct
  SOL       1x1             1408   struct
  p          1x6341          12682  char

>> COIL
COIL =
  name: 'No Model Selected'
  type: []

>>
  
```

**MARIE\_MAIN**

**Body Model**

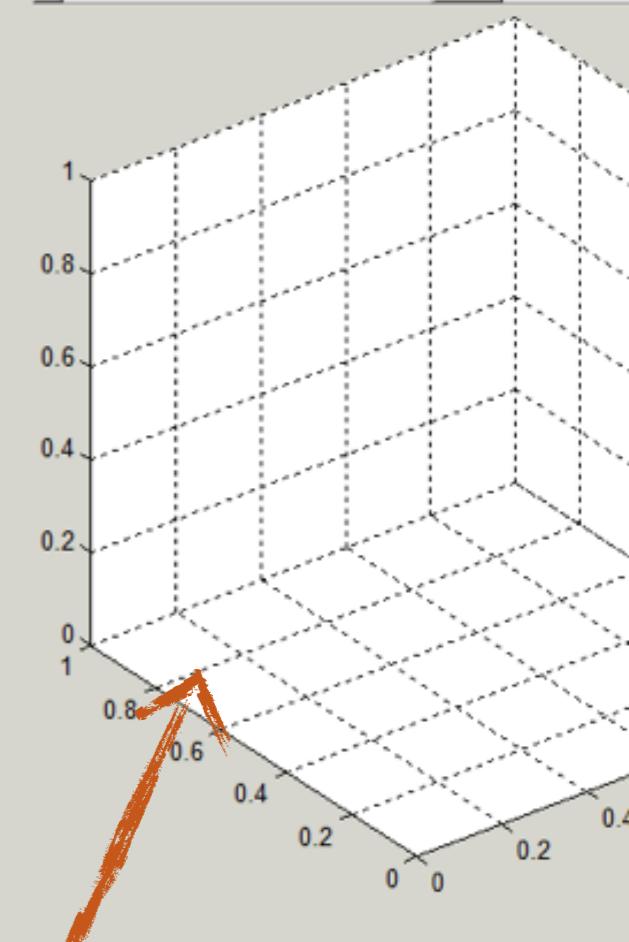
Model: No Model Select

**Coil Model**

Model: No Model Select

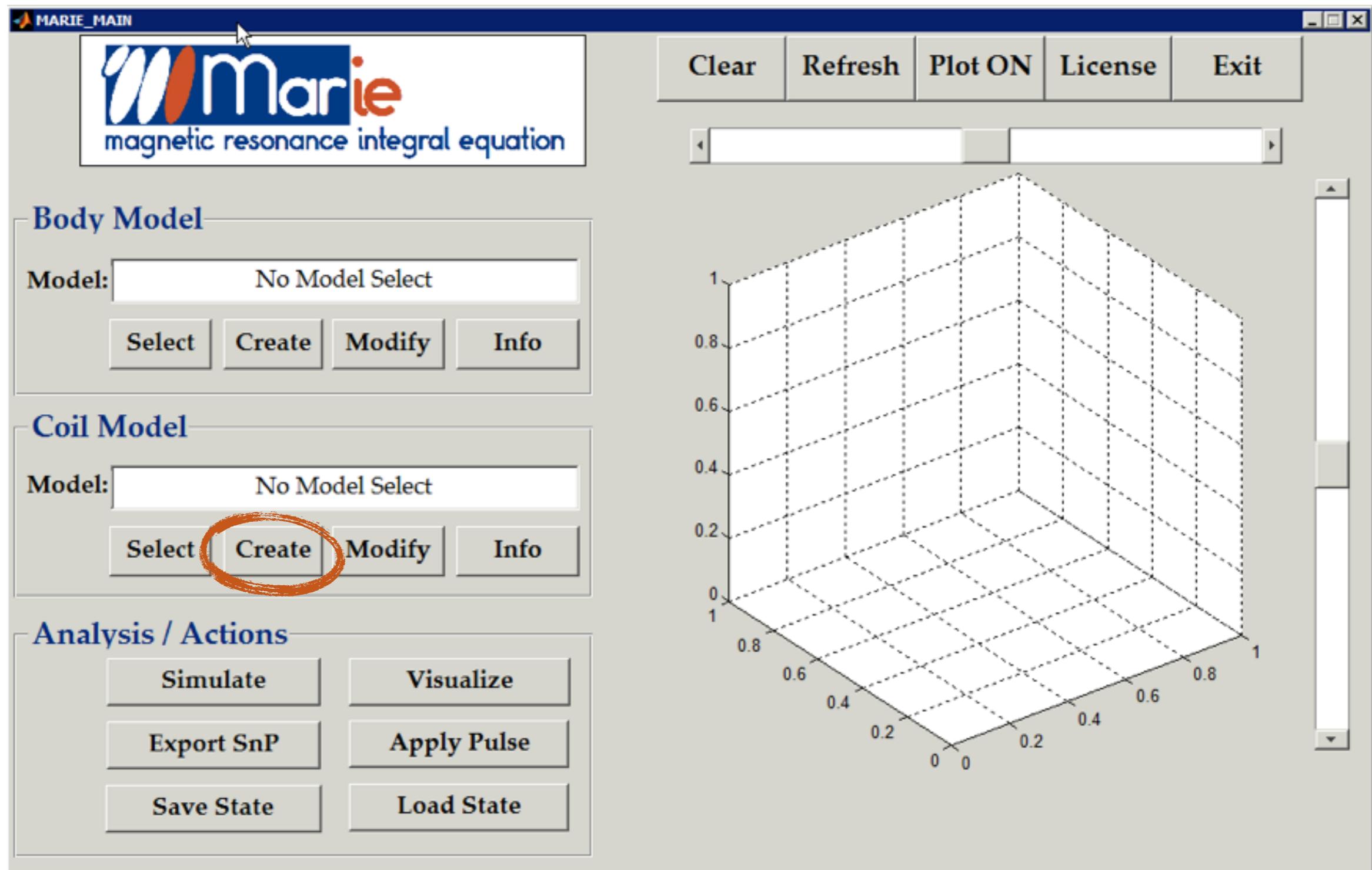
**Analysis / Actions**

**Clear Refresh Plot ON Lic**



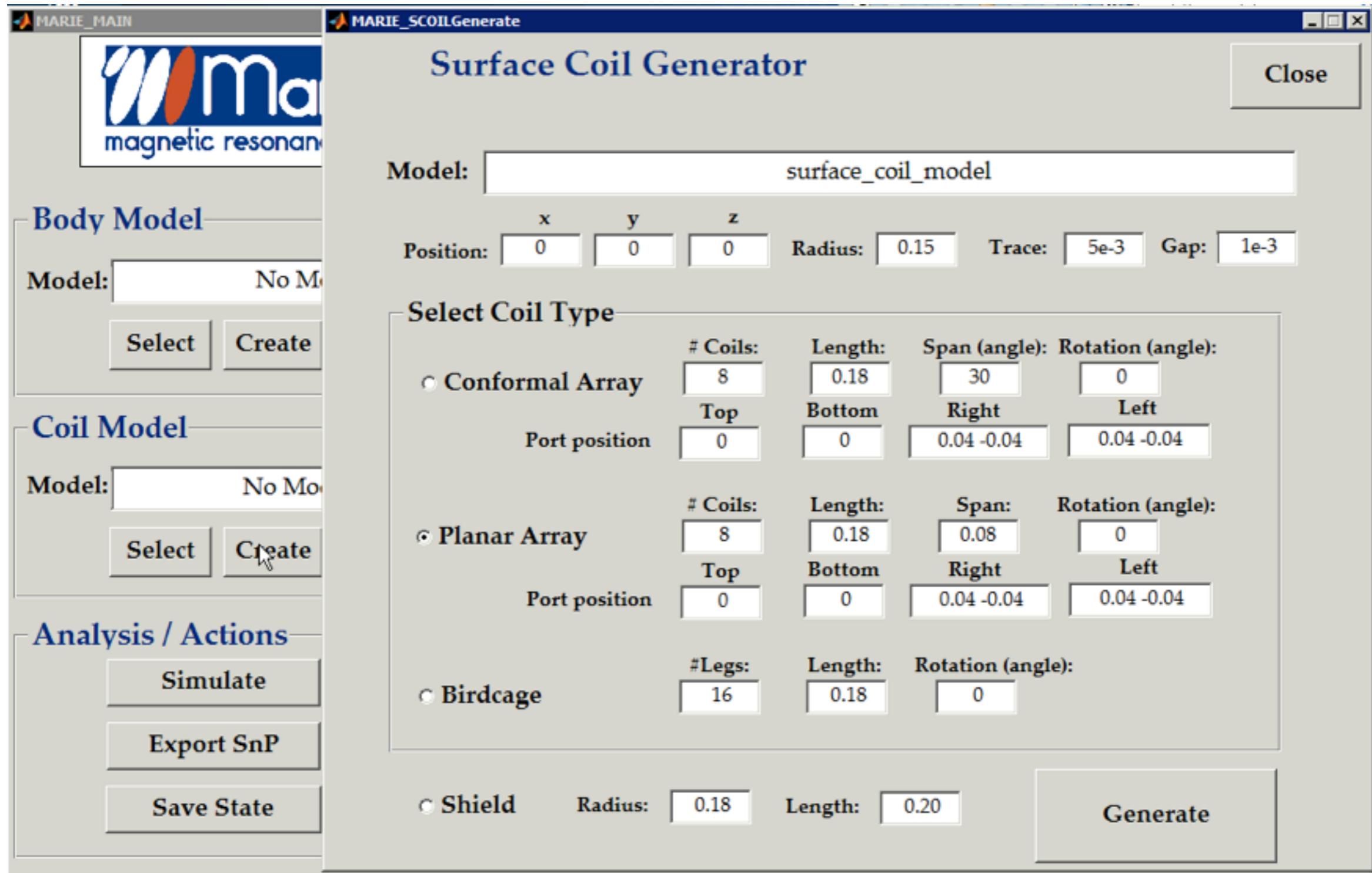
- ♦ Create a coil model

- ♦ push the create coil to call the coil wizard generator



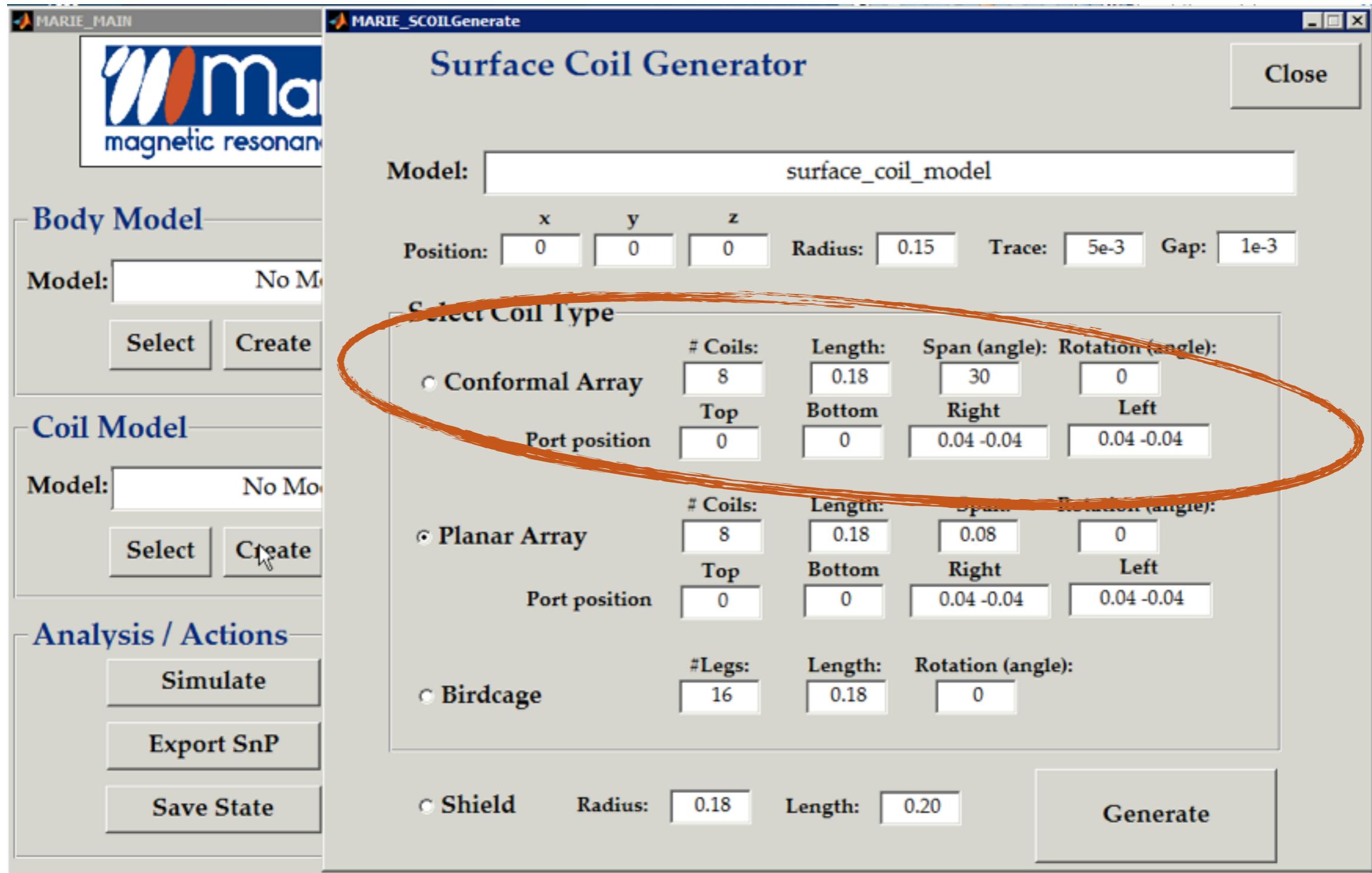
## ◆ Create a coil model

- ◆ push the create coil to call the coil wizard generator



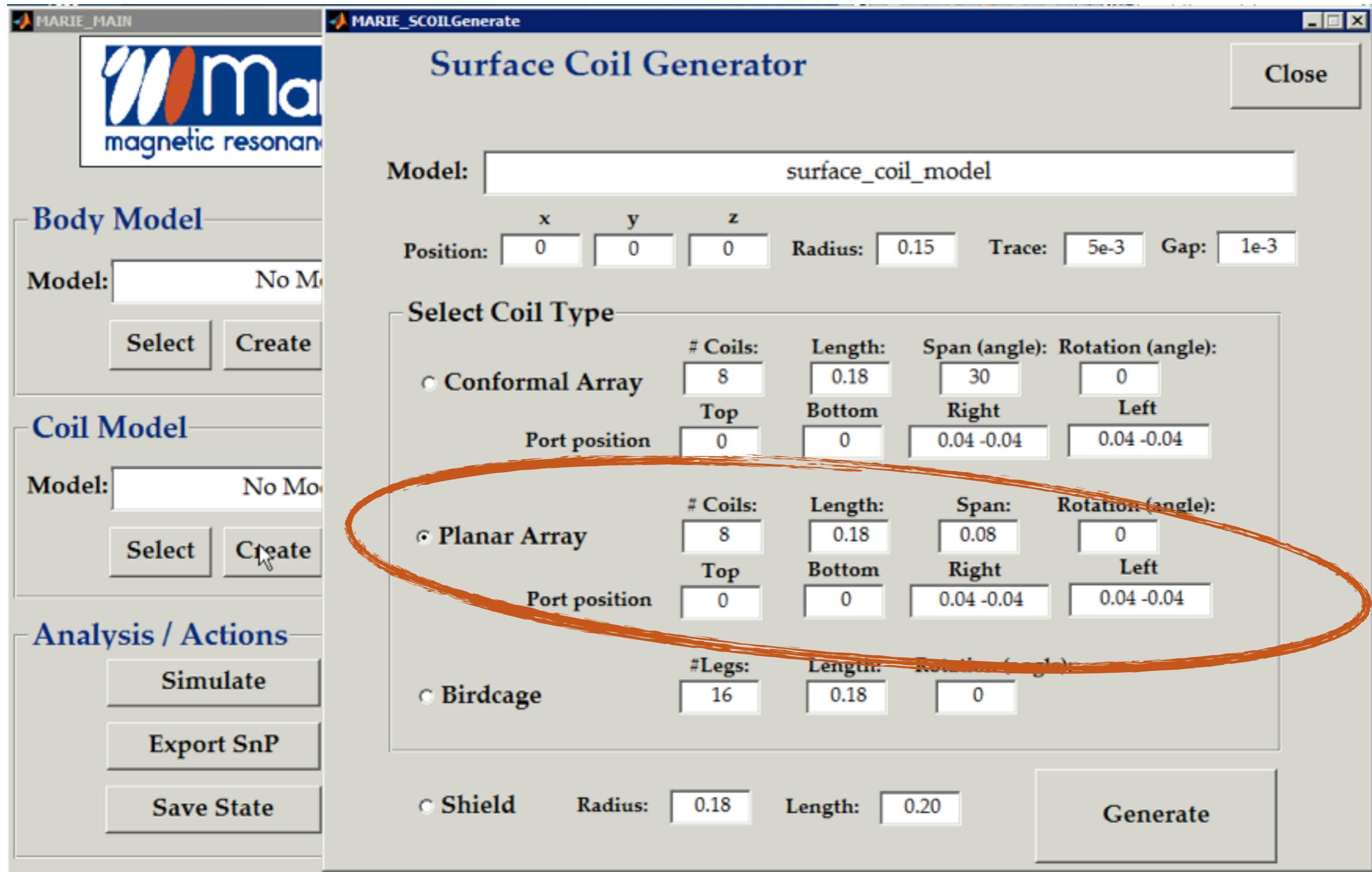
## ♦ Create a coil model

- ♦ Allows to automatically generate conformal coil arrays

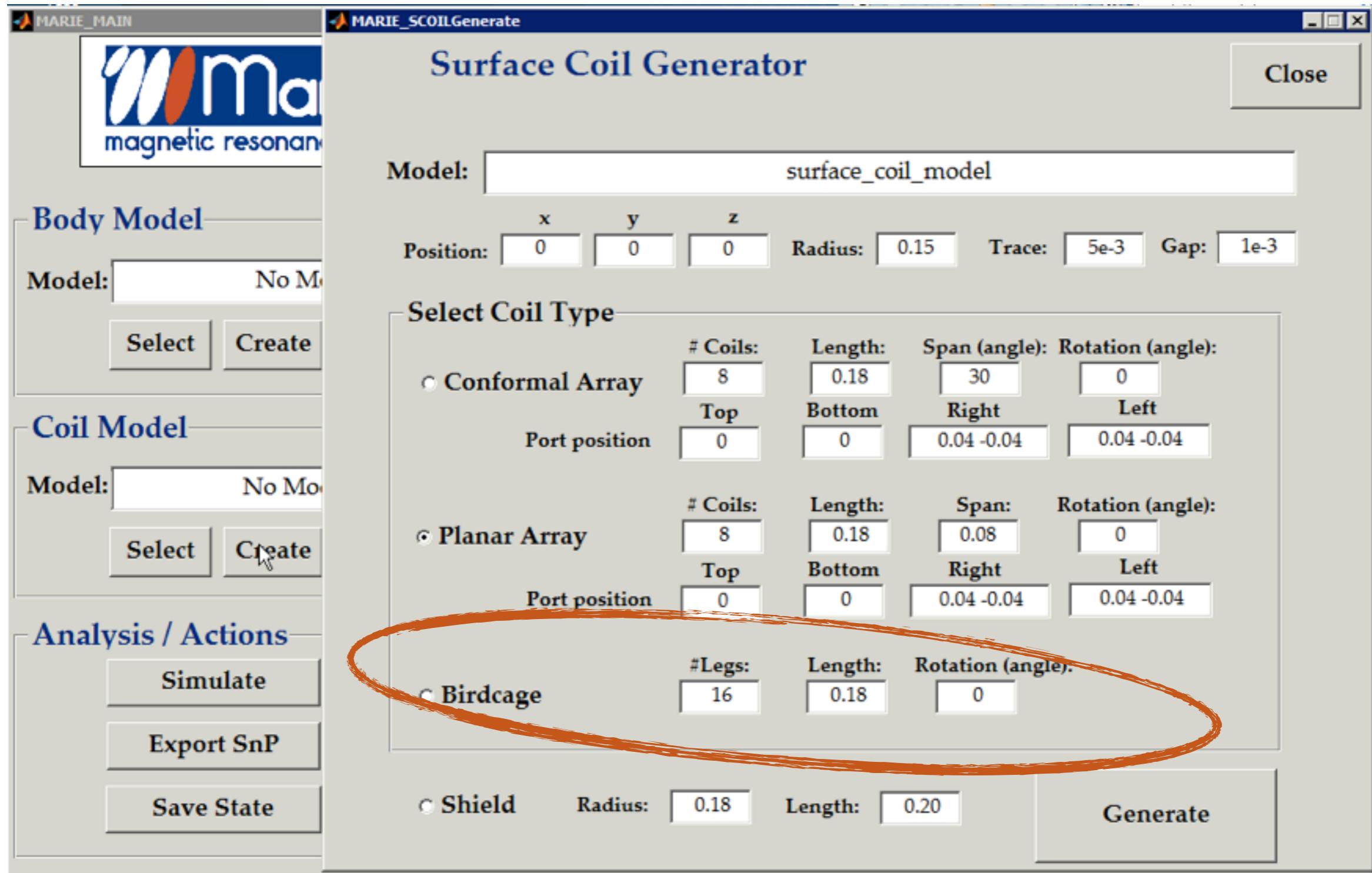


## ♦ Create a coil model

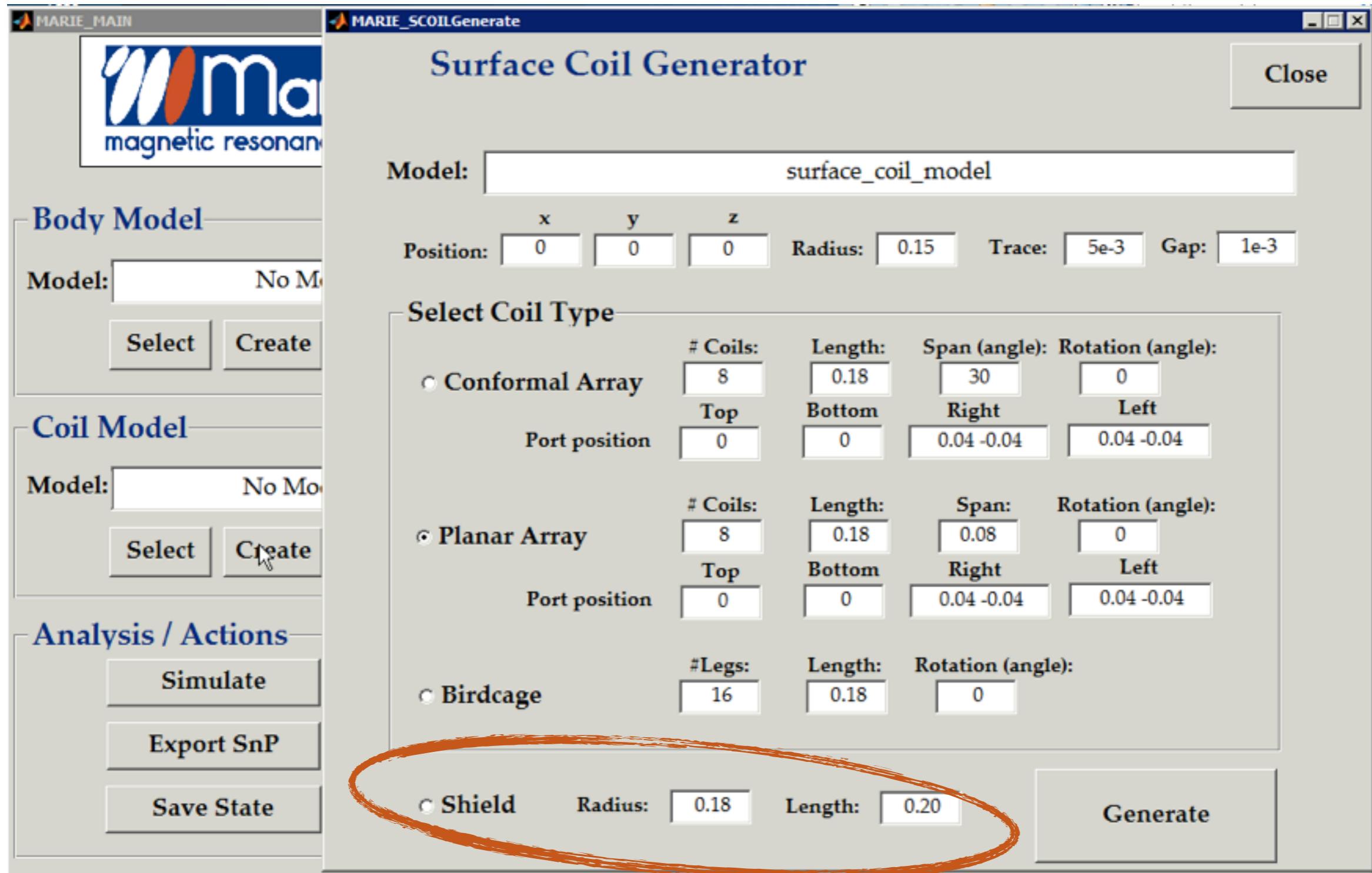
- ♦ Allows to automatically generate conformal coil arrays



- ◆ Create a coil model
  - ◆ Allows to automatically generate birdcage coils

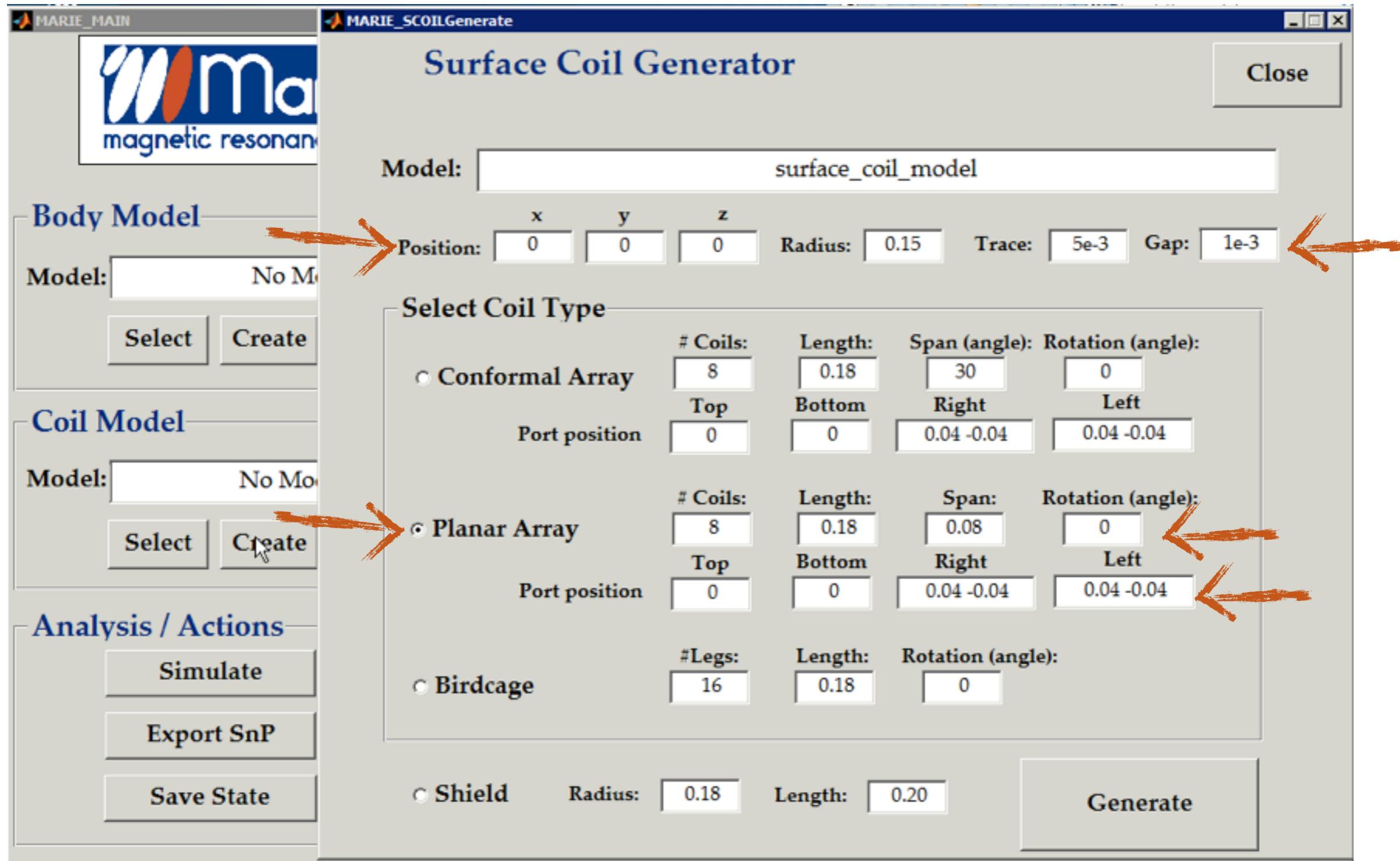


- ◆ Create a coil model
  - ◆ and also include shield



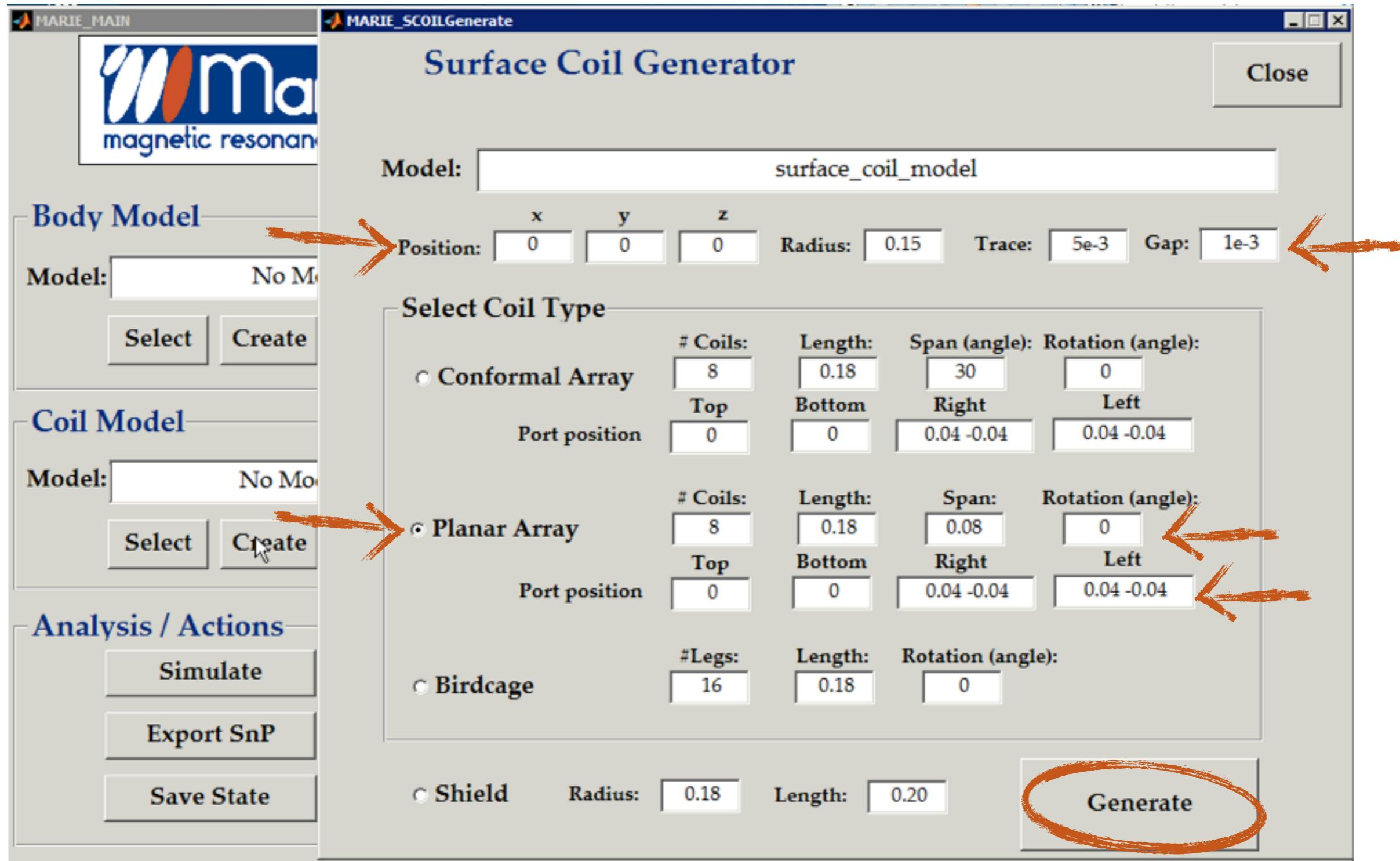
## ◆ Create a coil model

- ◆ select coil type, characteristics and position, and push generate



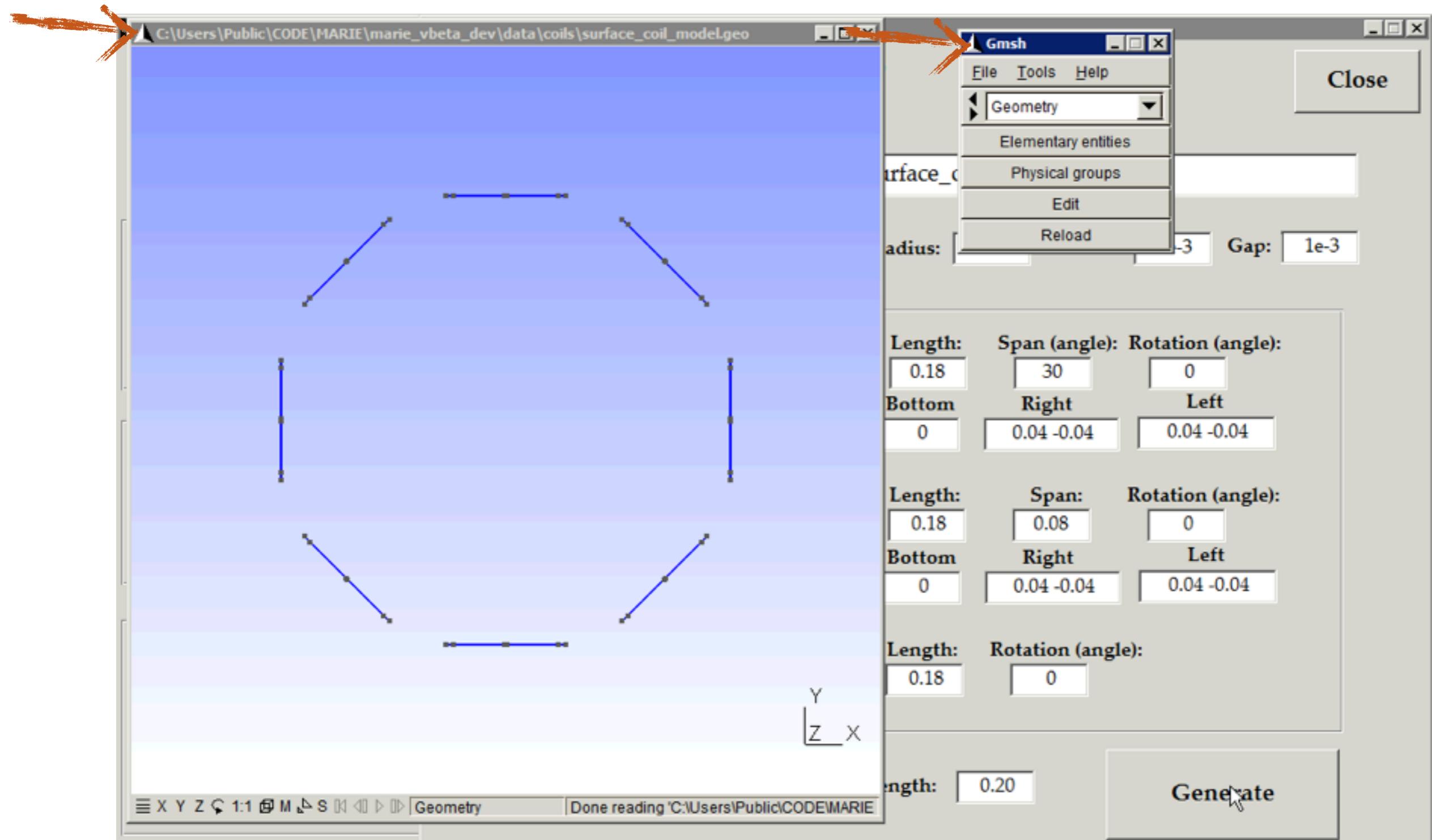
## ♦ Create a coil model

- ♦ Note: port positions are arrays of the position of the ports at each coil

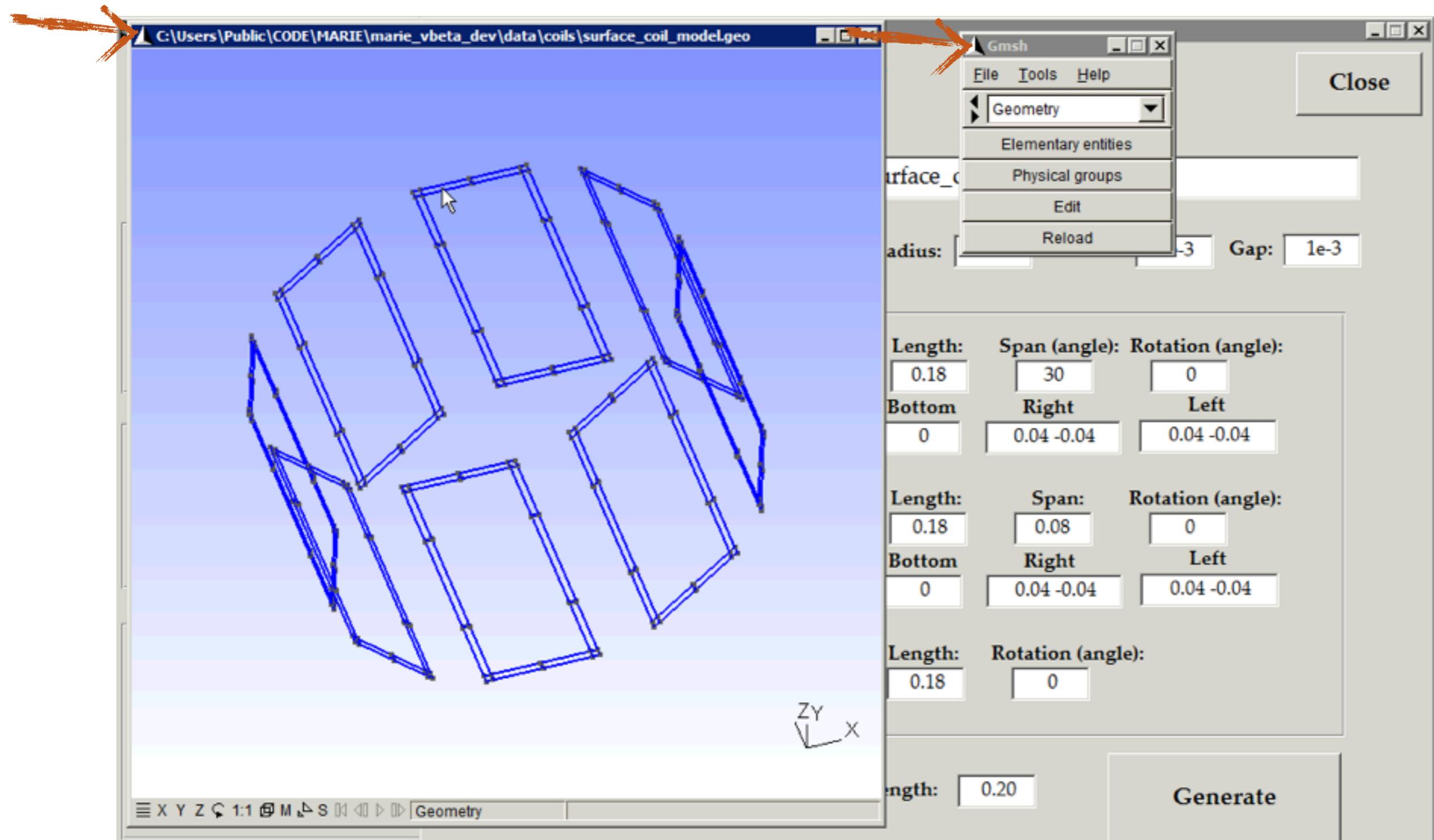


## ♦ Create a coil model

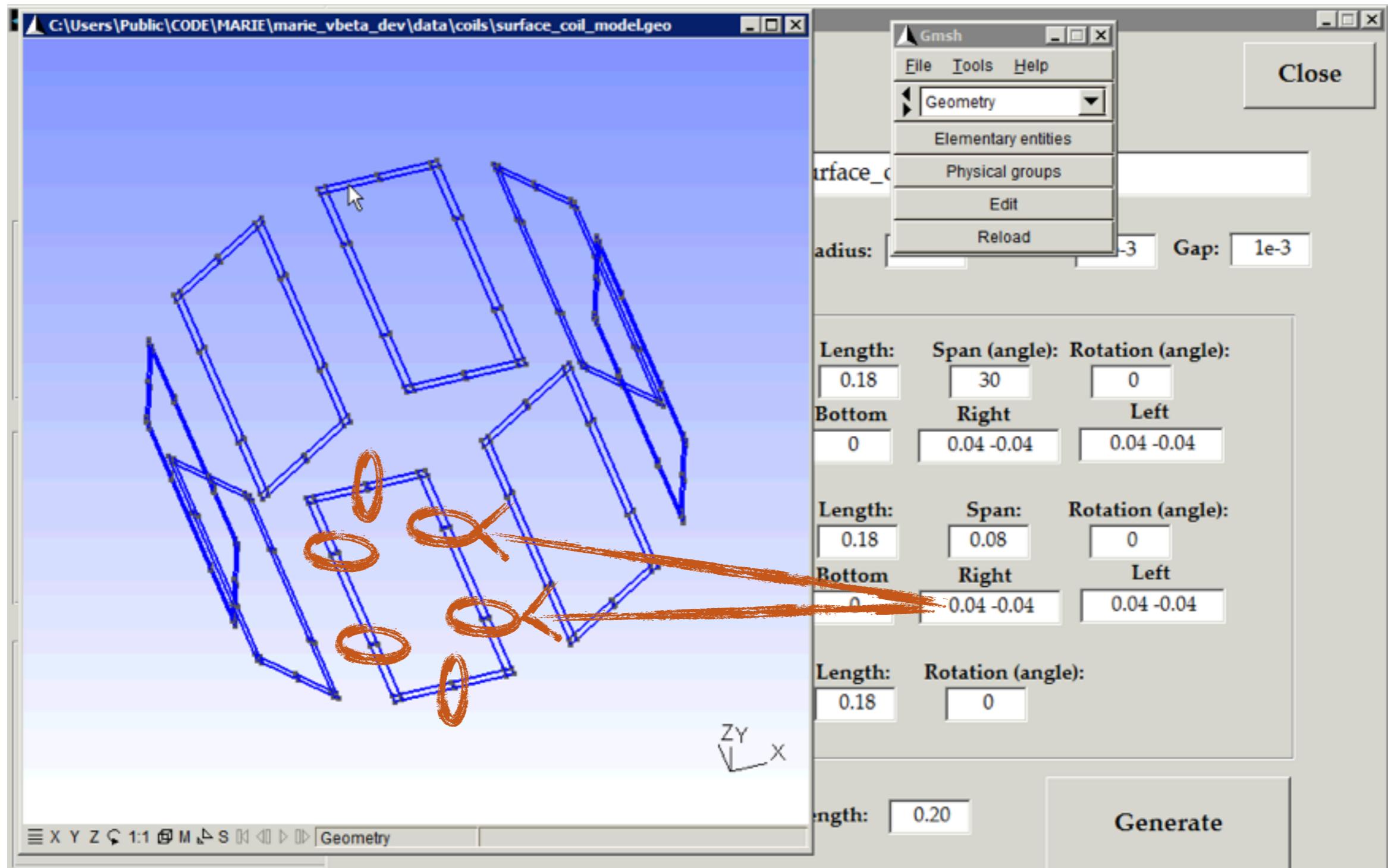
- ♦ will generate the .geo file and open an interface with gmsh with it



- ◆ Create a coil model
  - ◆ 3D visual environment to visualize the geometry

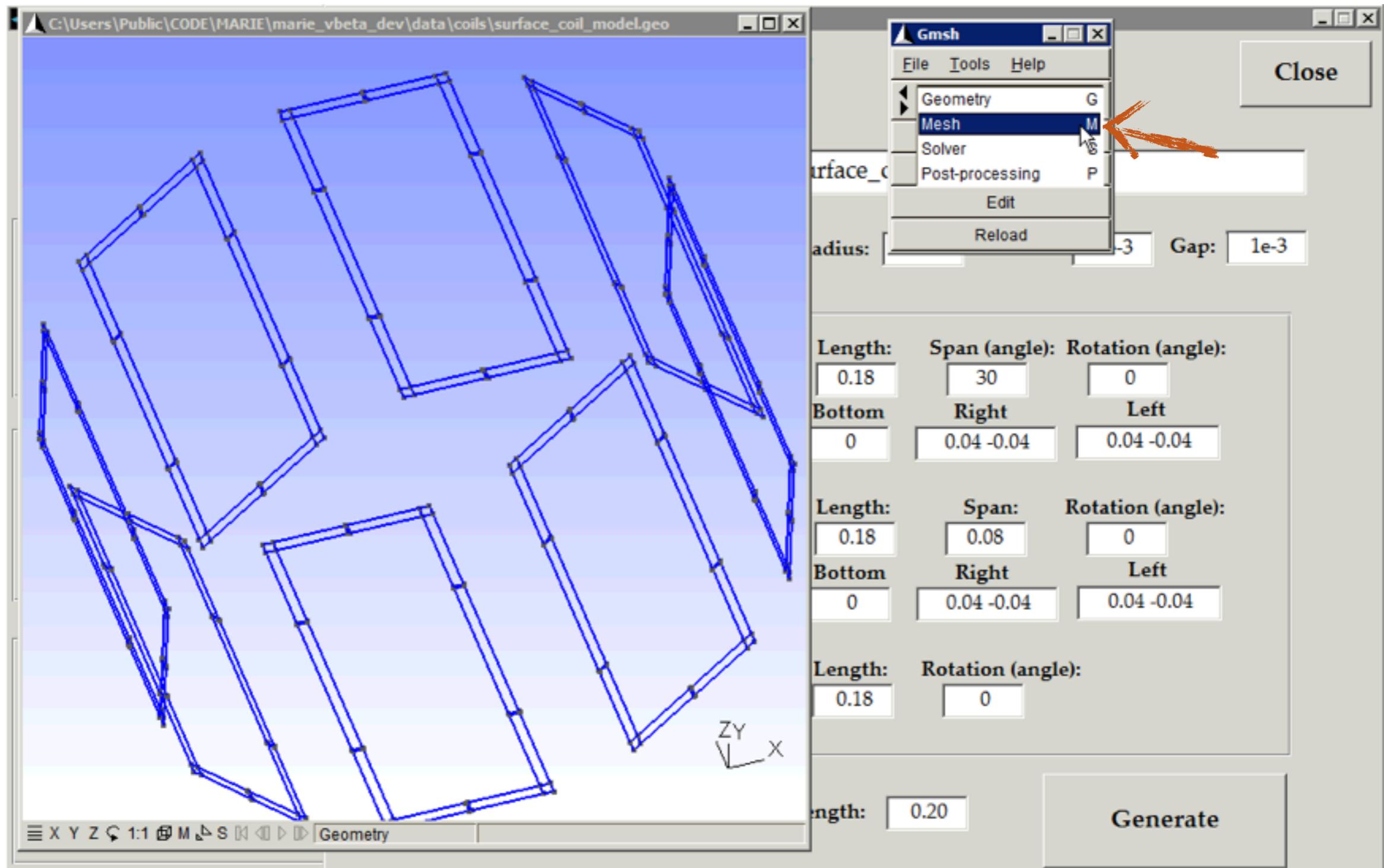


- ◆ Create a coil model
  - ◆ note ports (gaps) positions in each coil

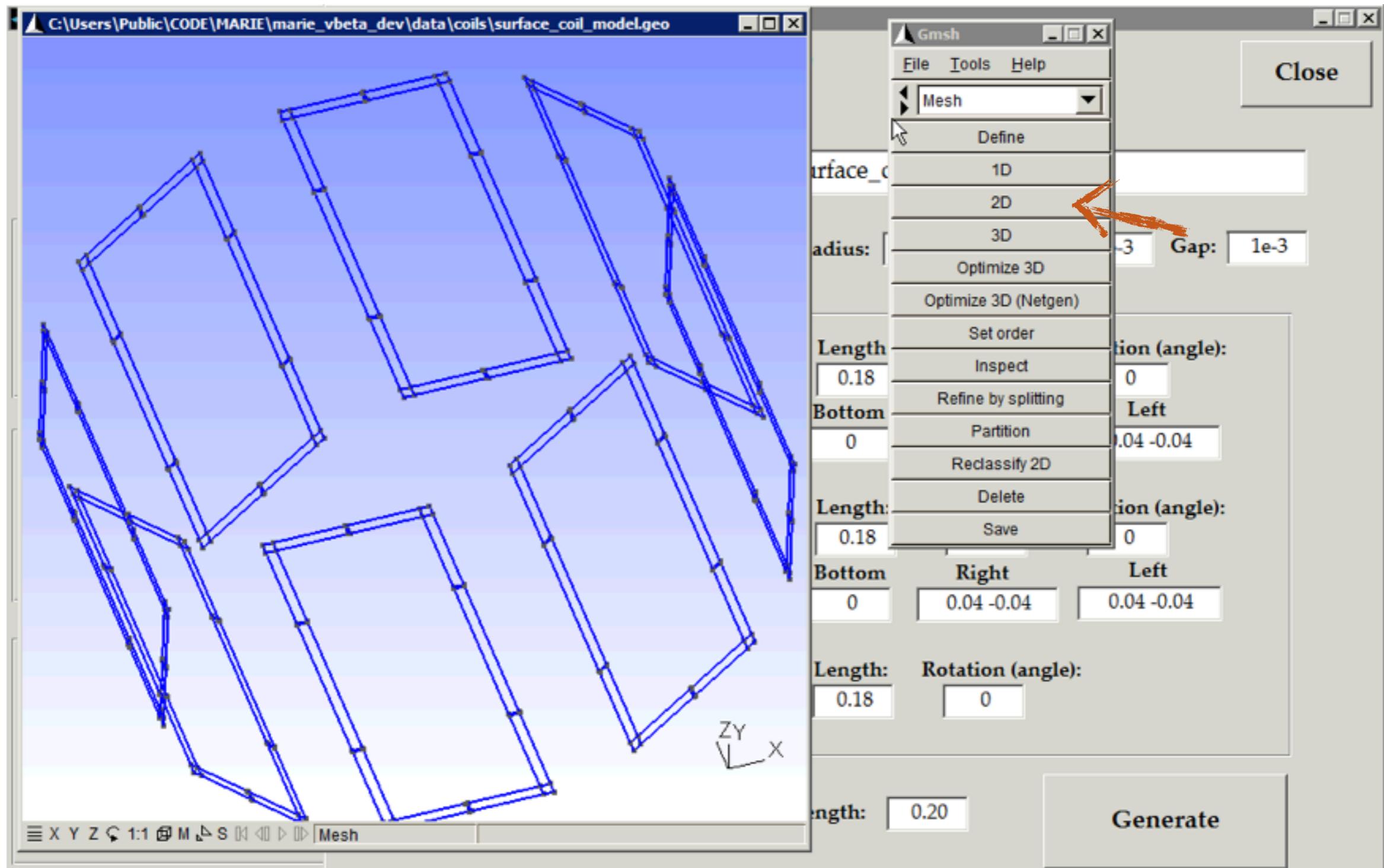


- ♦ Create a coil model

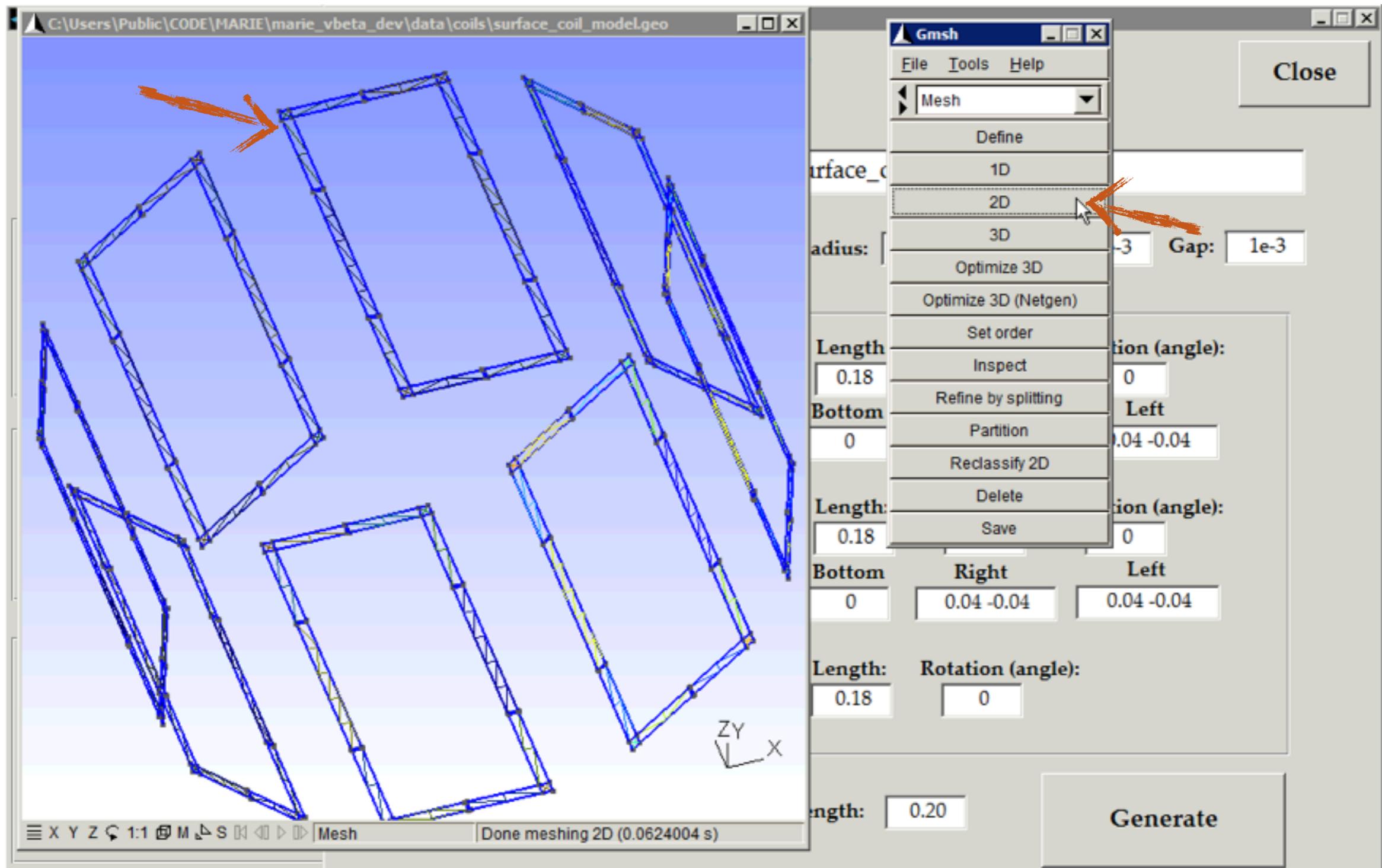
- ♦ allows you to automatically generate a triangular mesh from .geo



- ♦ Create a coil model
  - ♦ go to 2D mesh to perform the tessellation

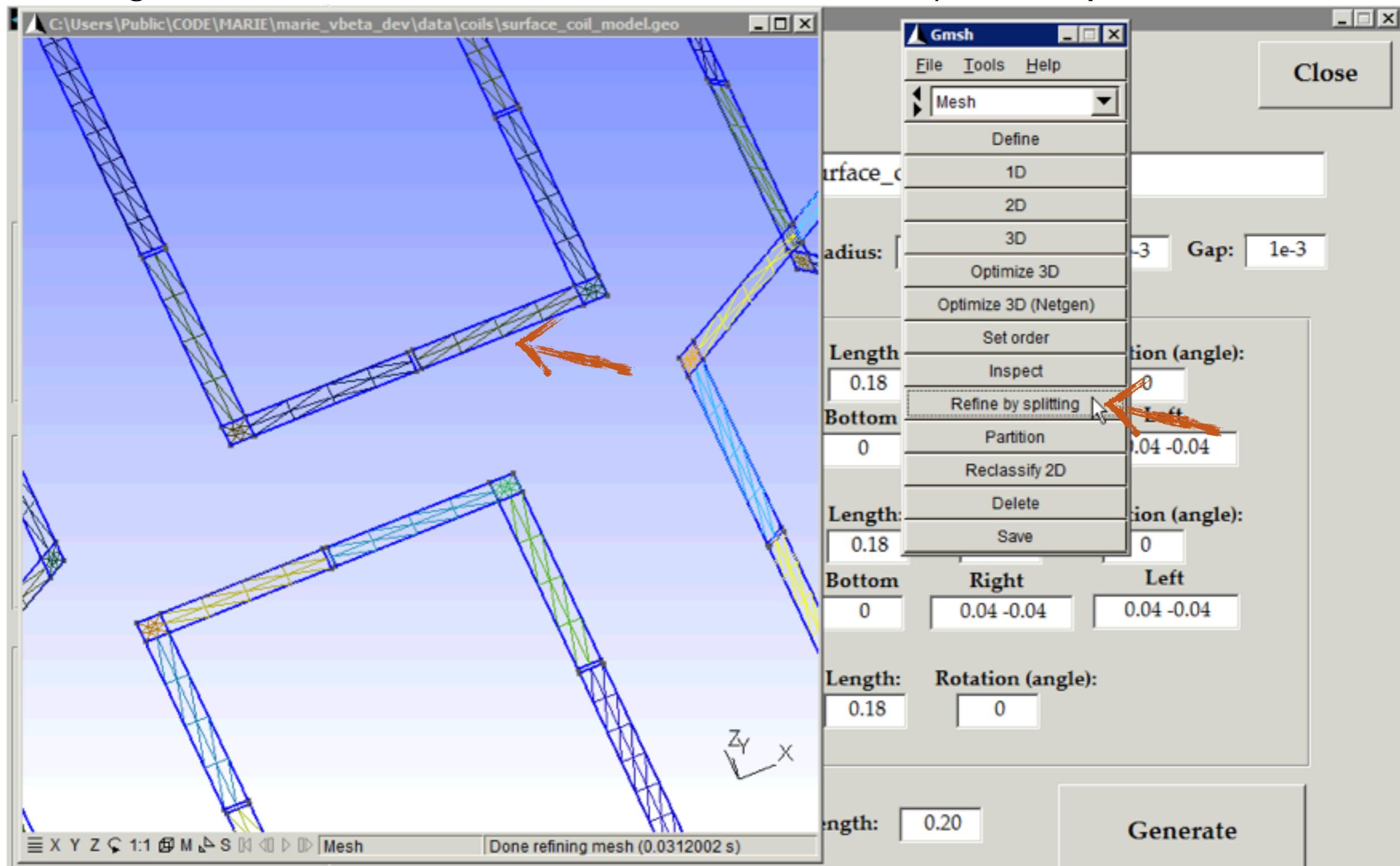


- ♦ Create a coil model
  - ♦ note the discretization in the view



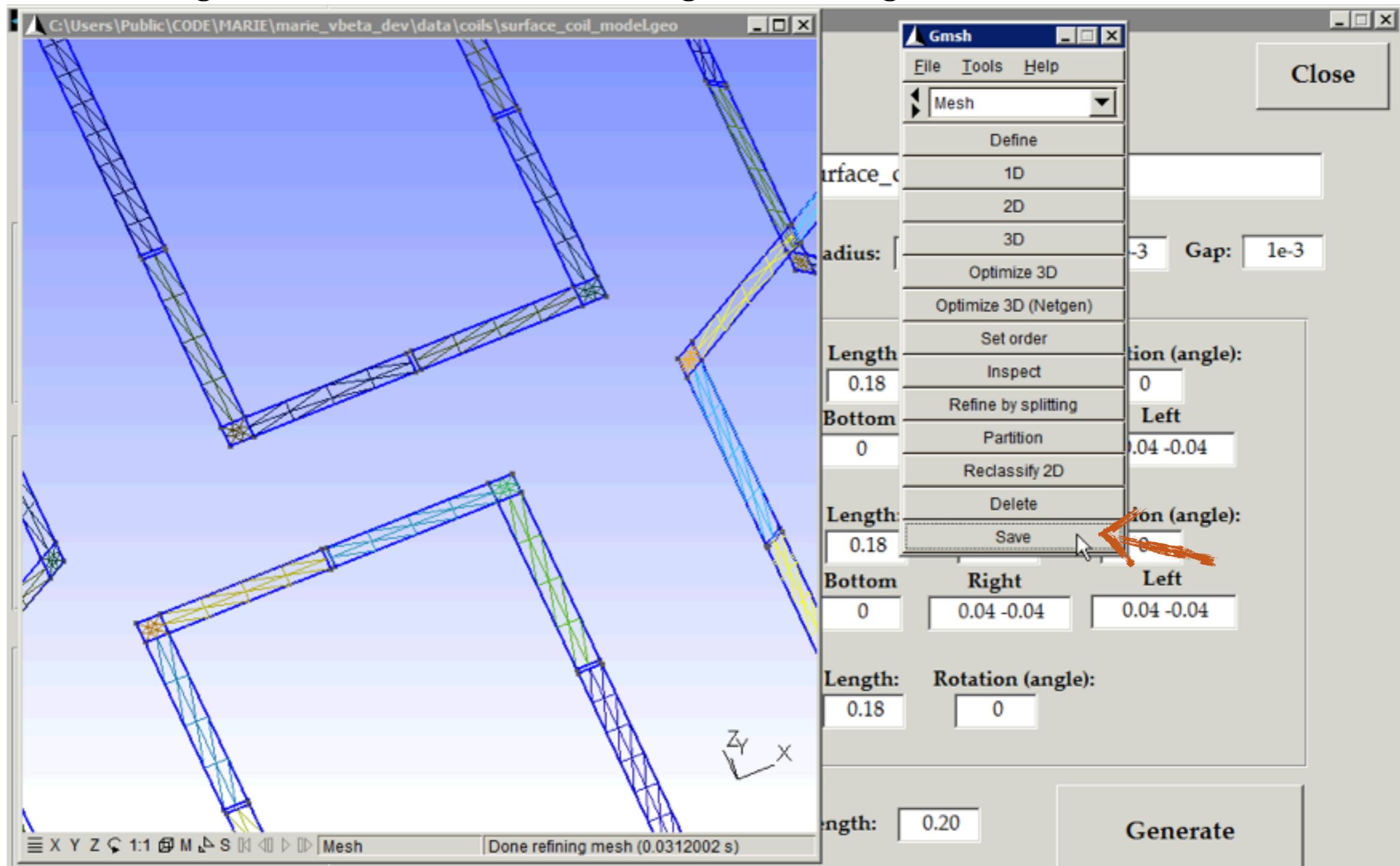
## ♦ Create a coil model

- ♦ .geo is prepared to generate a coarse discretization: refine by splitting
  - ♦ to generate a finer discretization (better accuracy at computational cost)



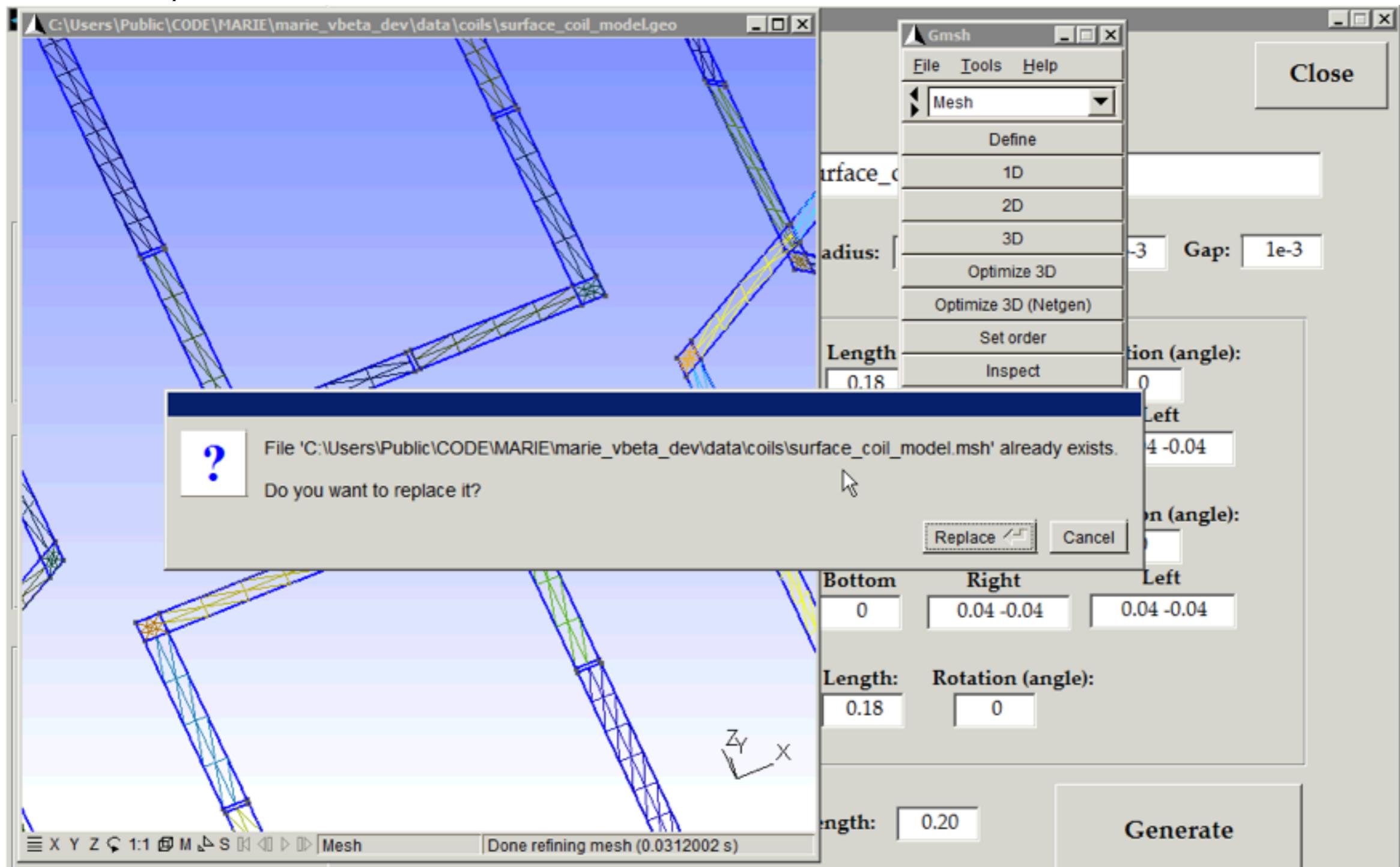
## ♦ Create a coil model

- ♦ push save to save the corresponding files
  - ♦ it will generate the .msh file, along with the .geo and .smm

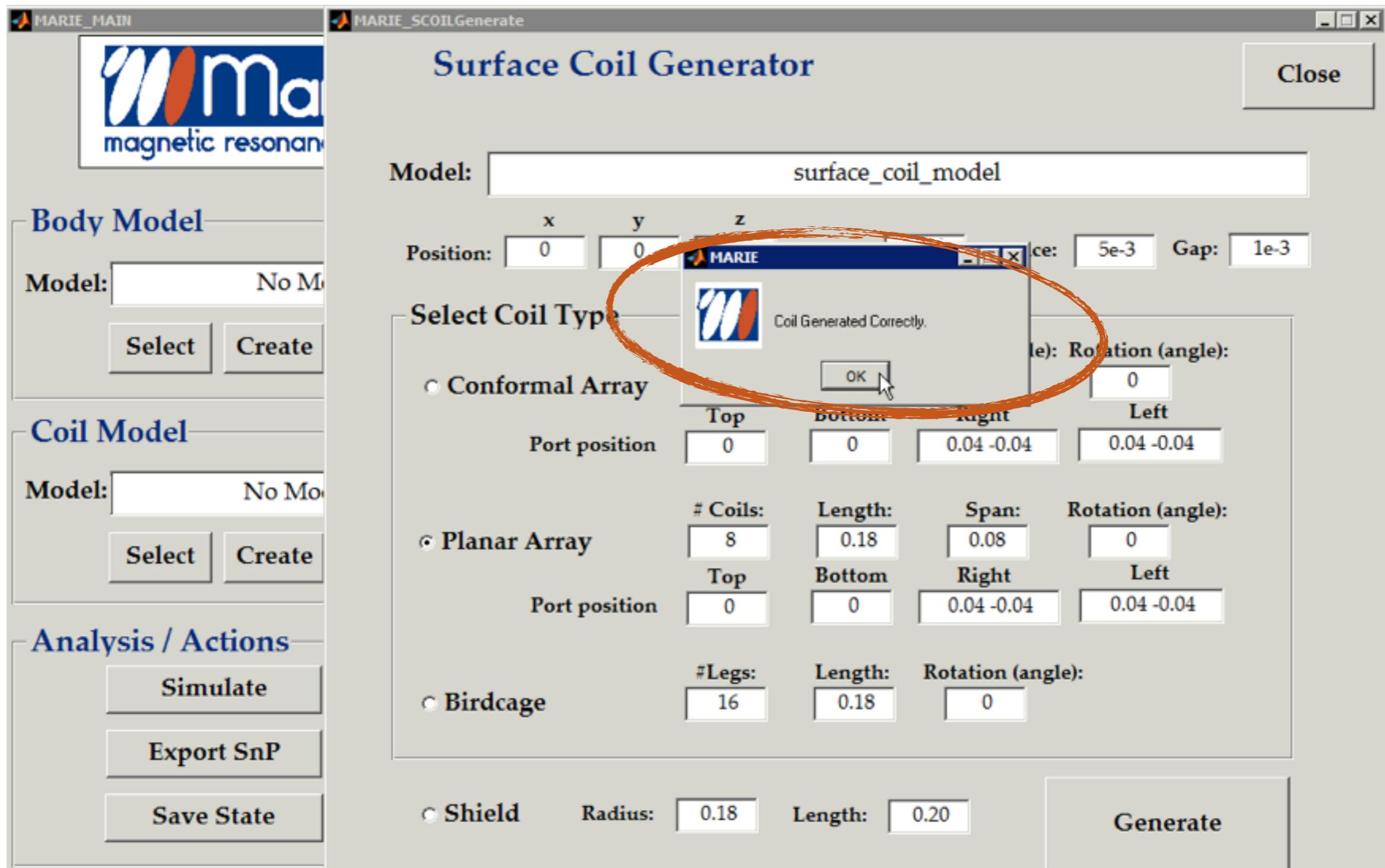


## ♦ Create a coil model

- ♦ push save to save the corresponding files
  - ♦ if they exist, will be overwritten

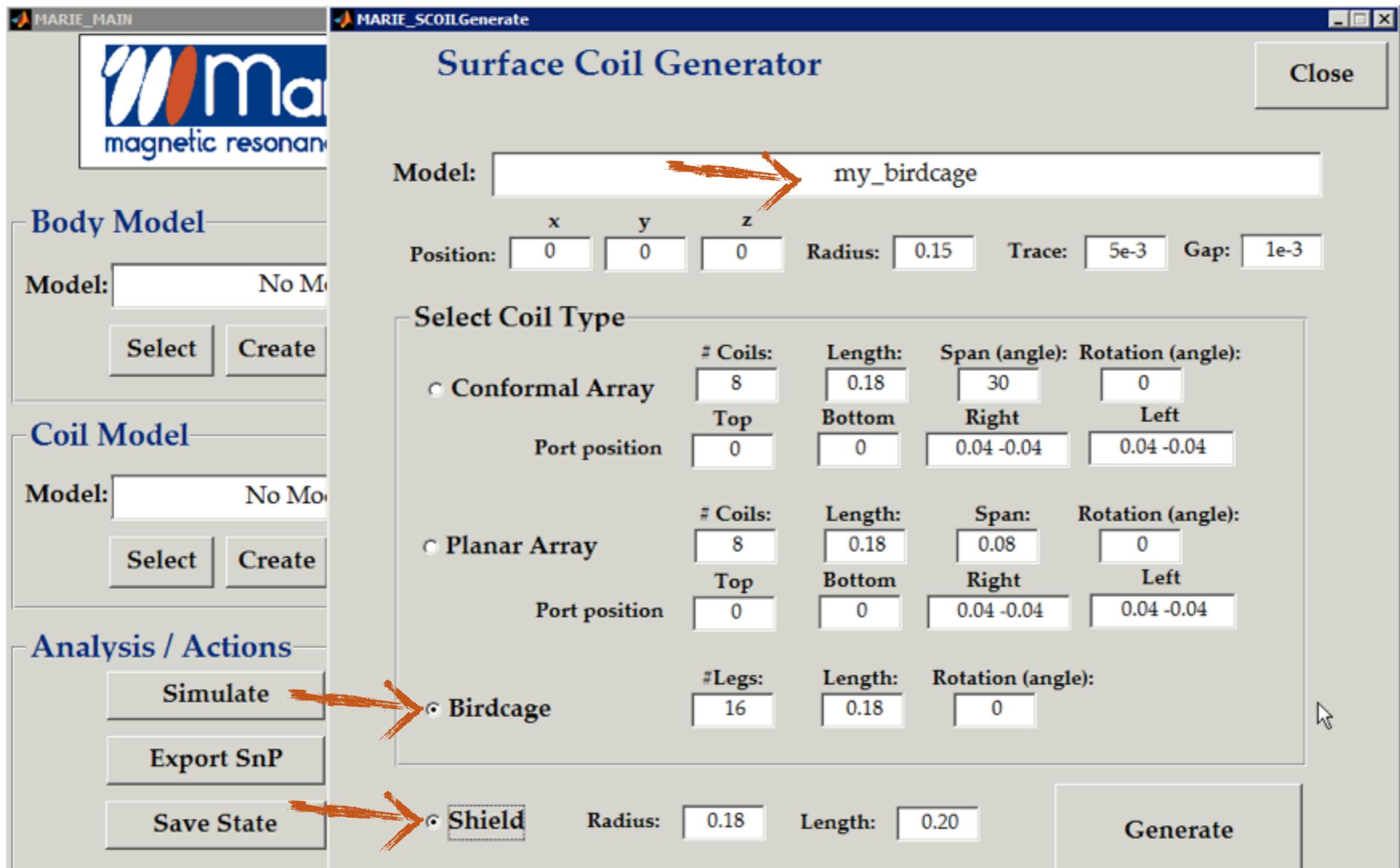


- ◆ Create a coil model
  - ◆ a confirmation will be issued

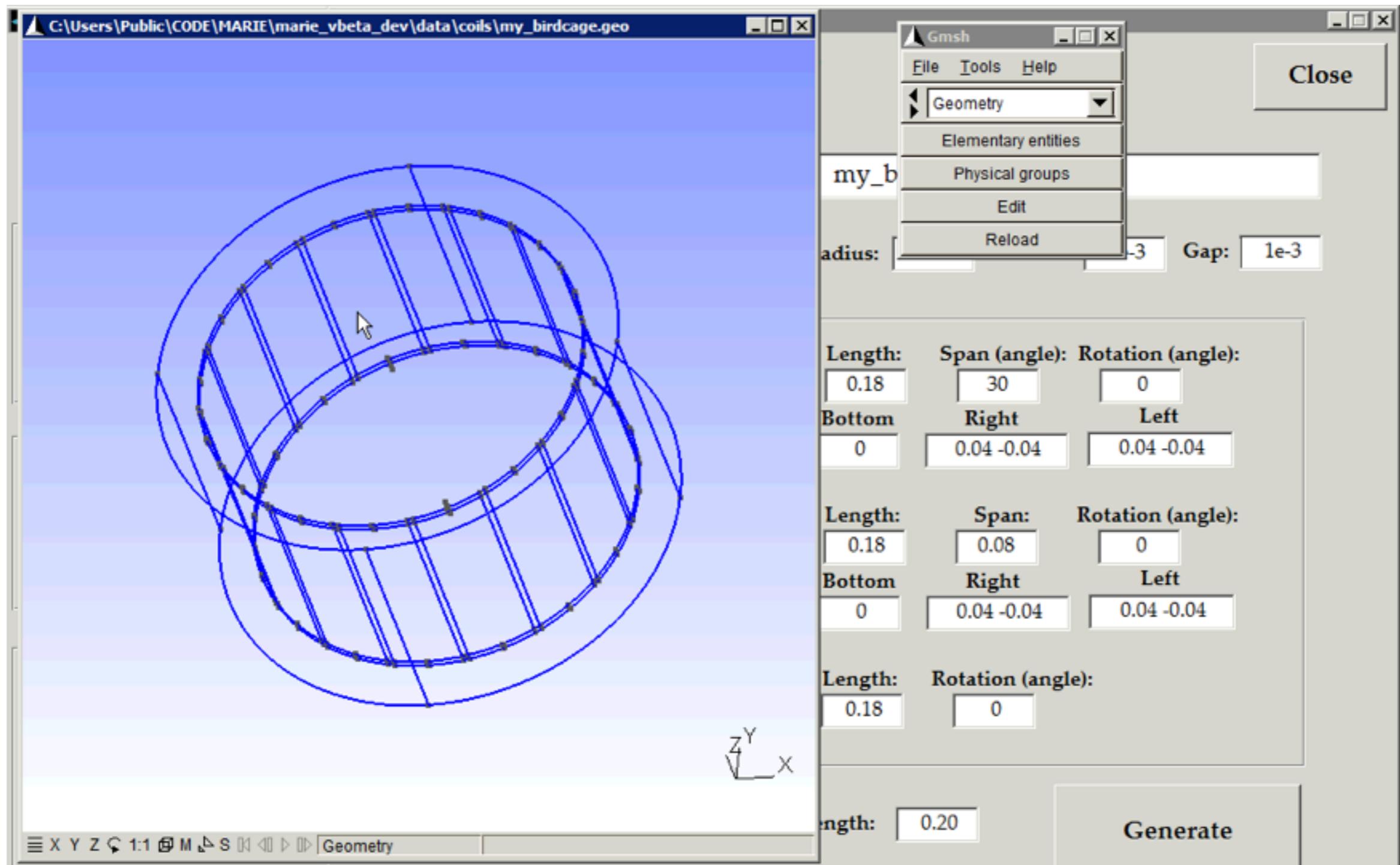


## ♦ Create a coil model

- ♦ a different geometry can be selected, including shield

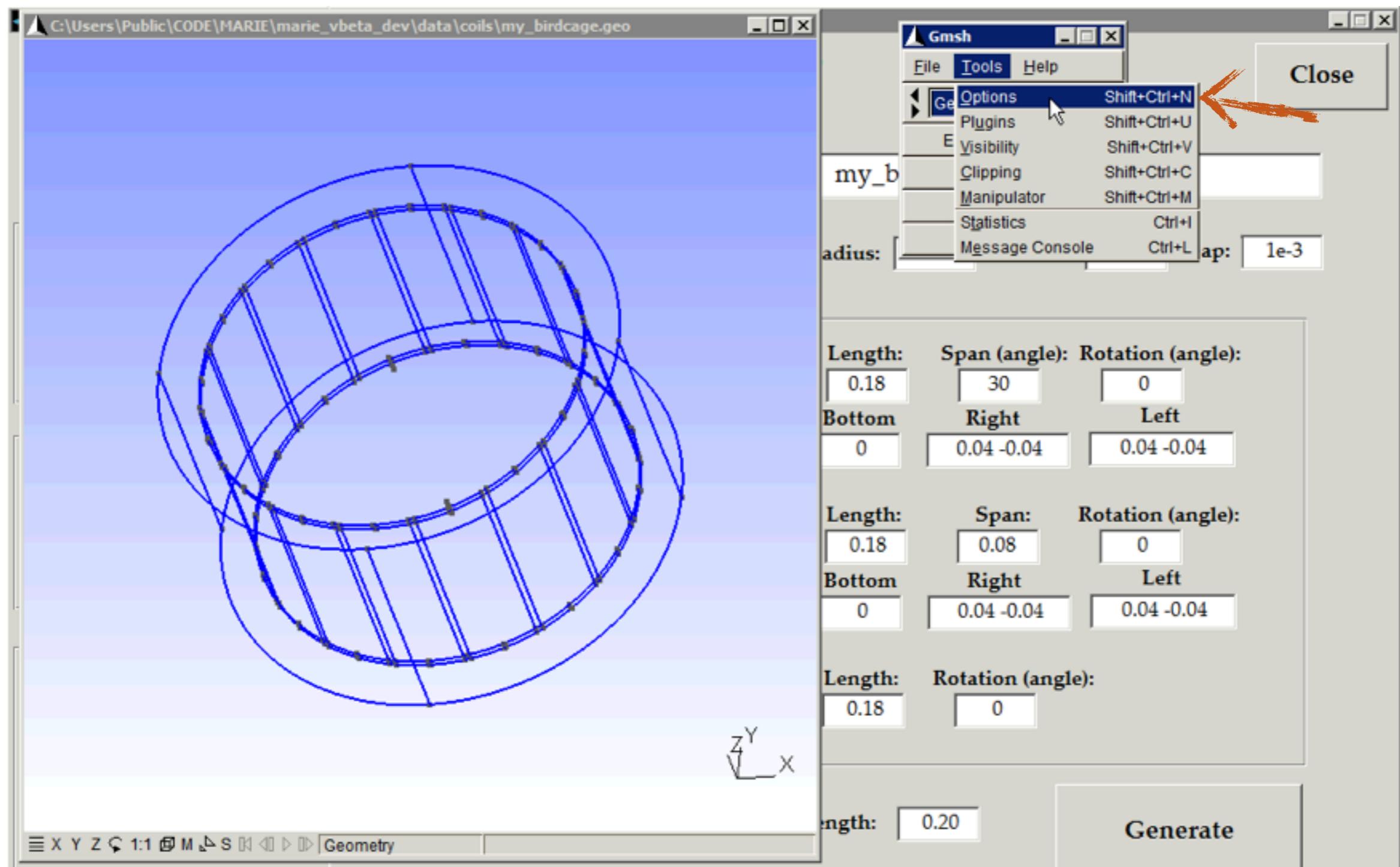


- ◆ Create a coil model
  - ◆ same gmsh environment



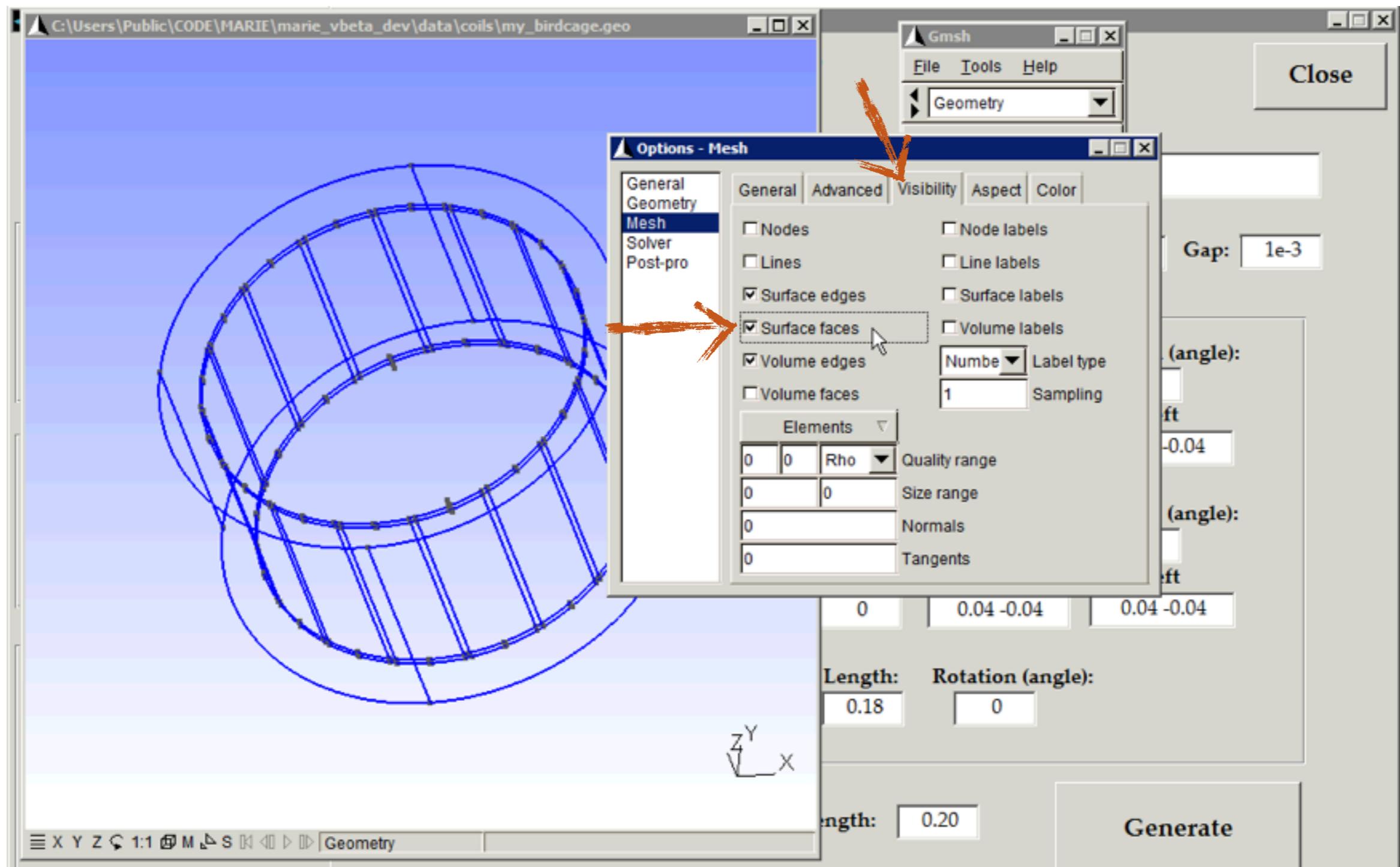
## ♦ Create a coil model

- ♦ we can modify the visibility options to show the geometry clearer

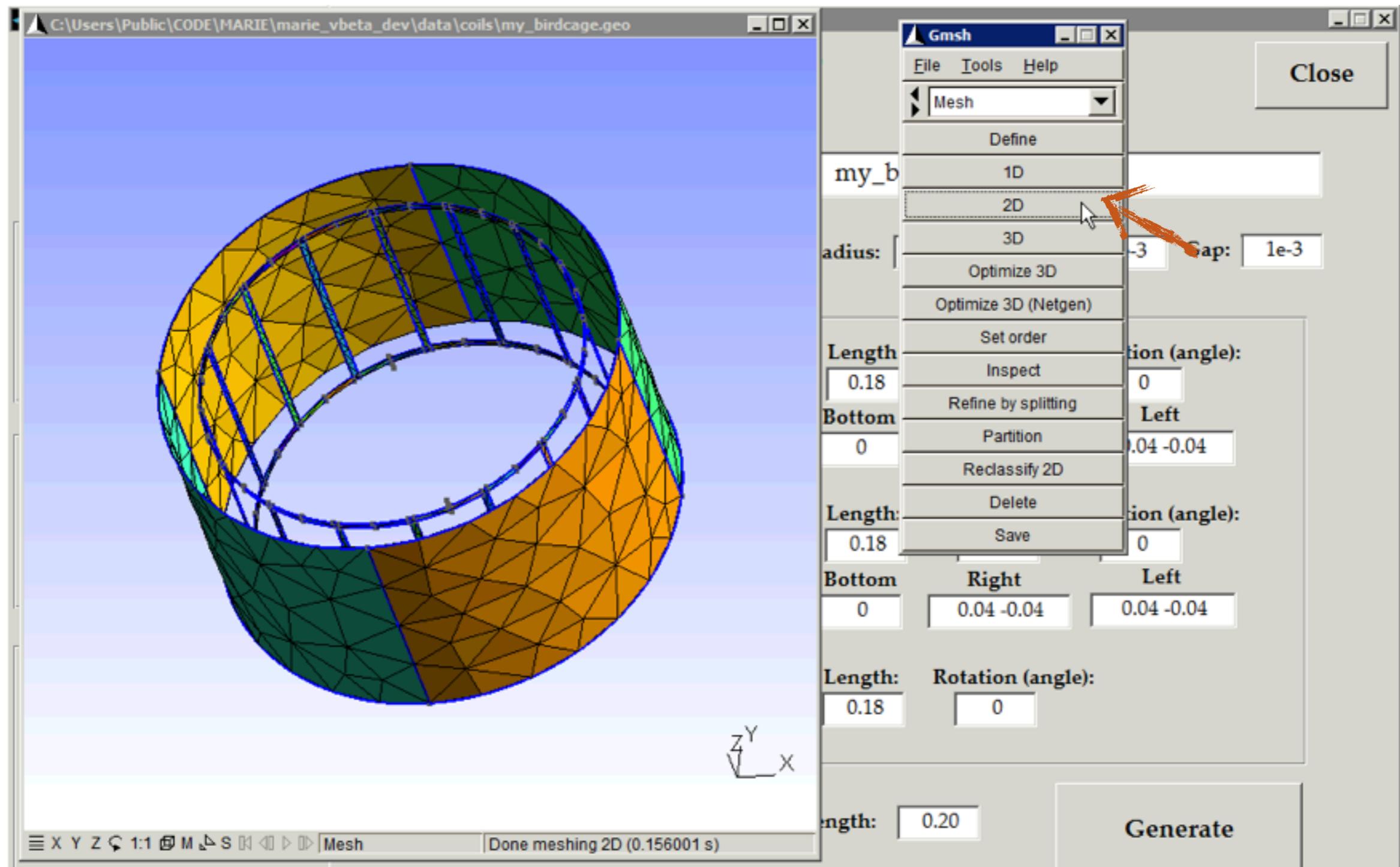


## ♦ Create a coil model

- ♦ we can modify the visibility options to show the geometry clearer

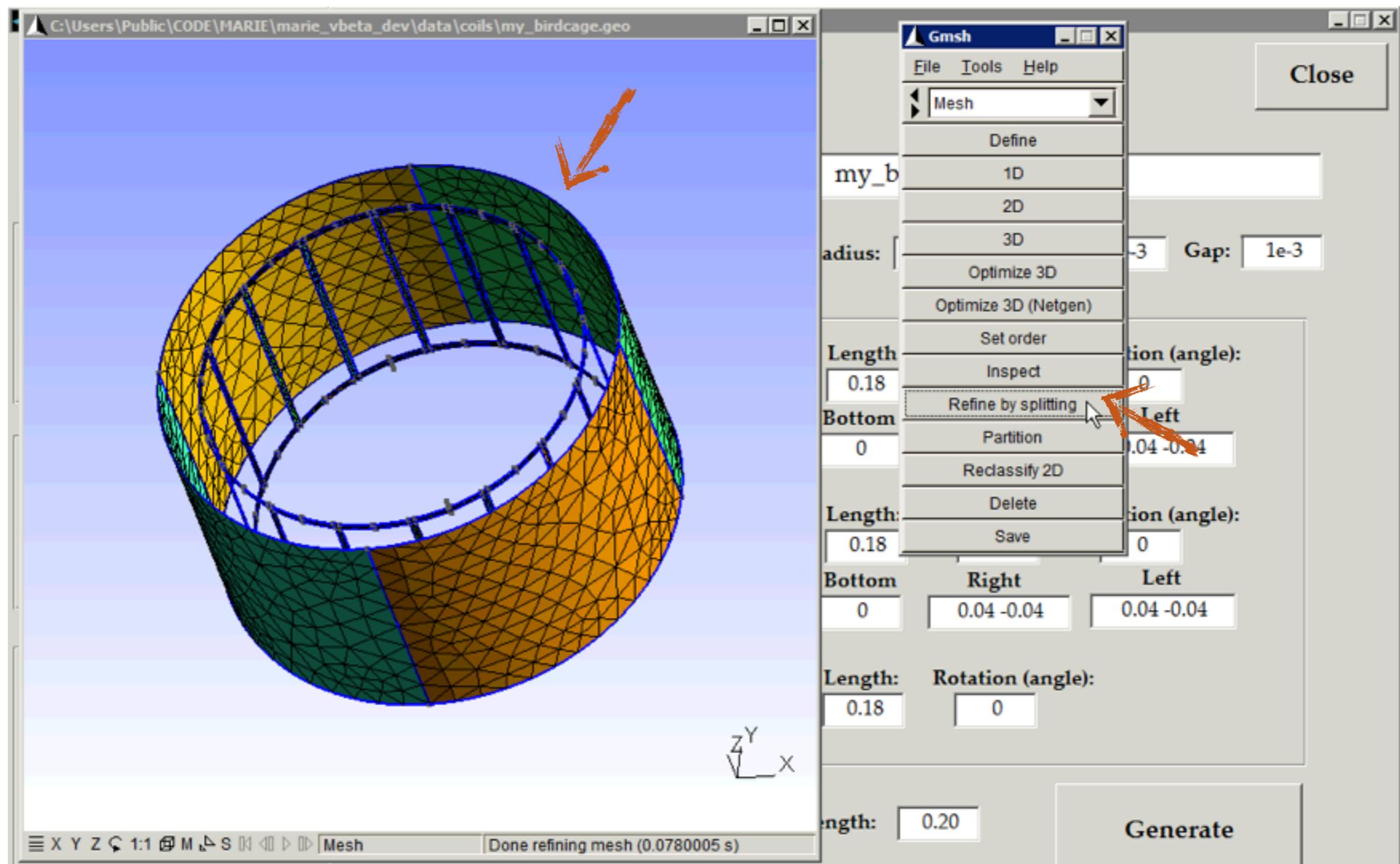


- ♦ Create a coil model
  - ♦ and the discretization



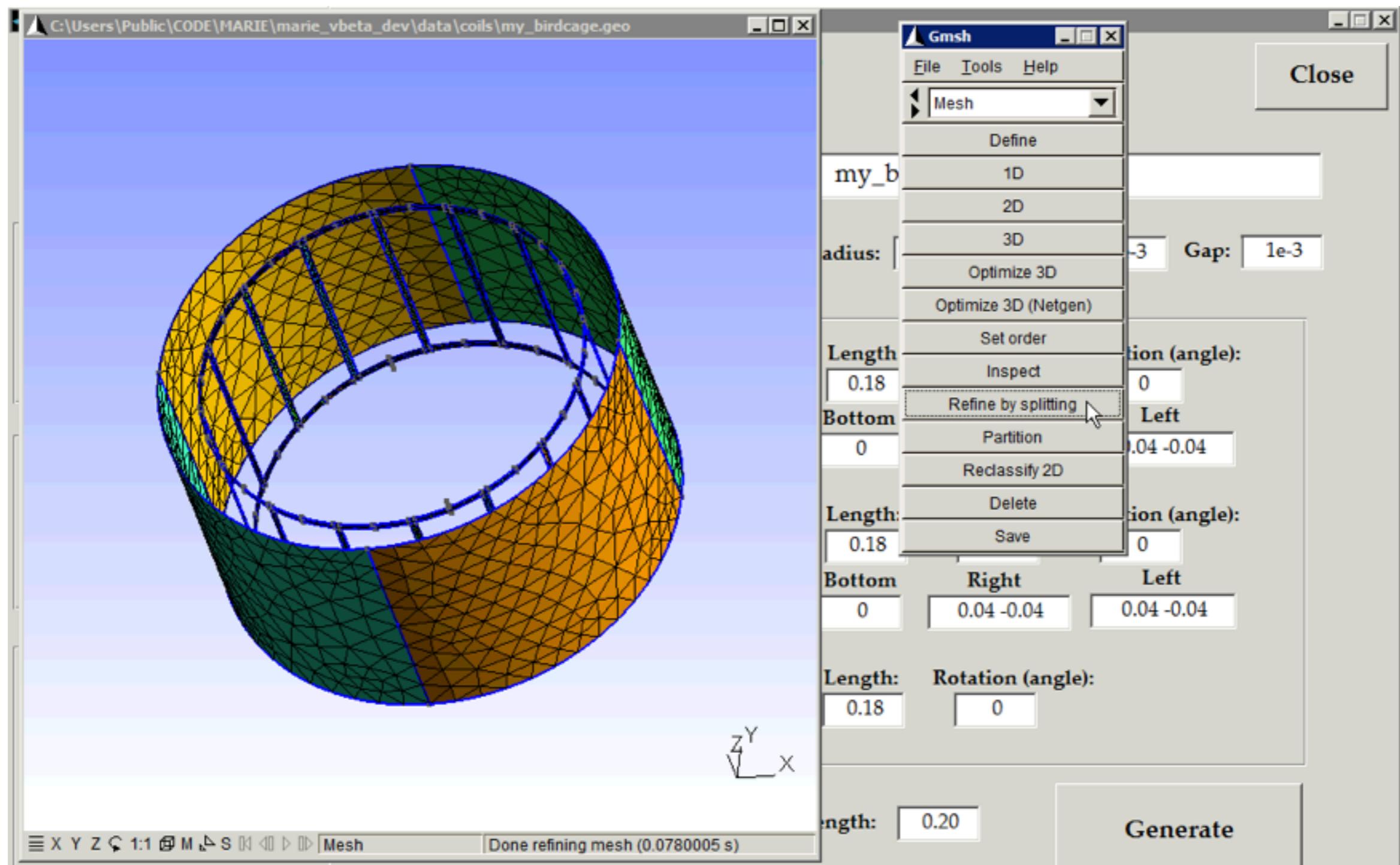
- ♦ Create a coil model

- ♦ again, refine by splitting to generate finer regular tessellation



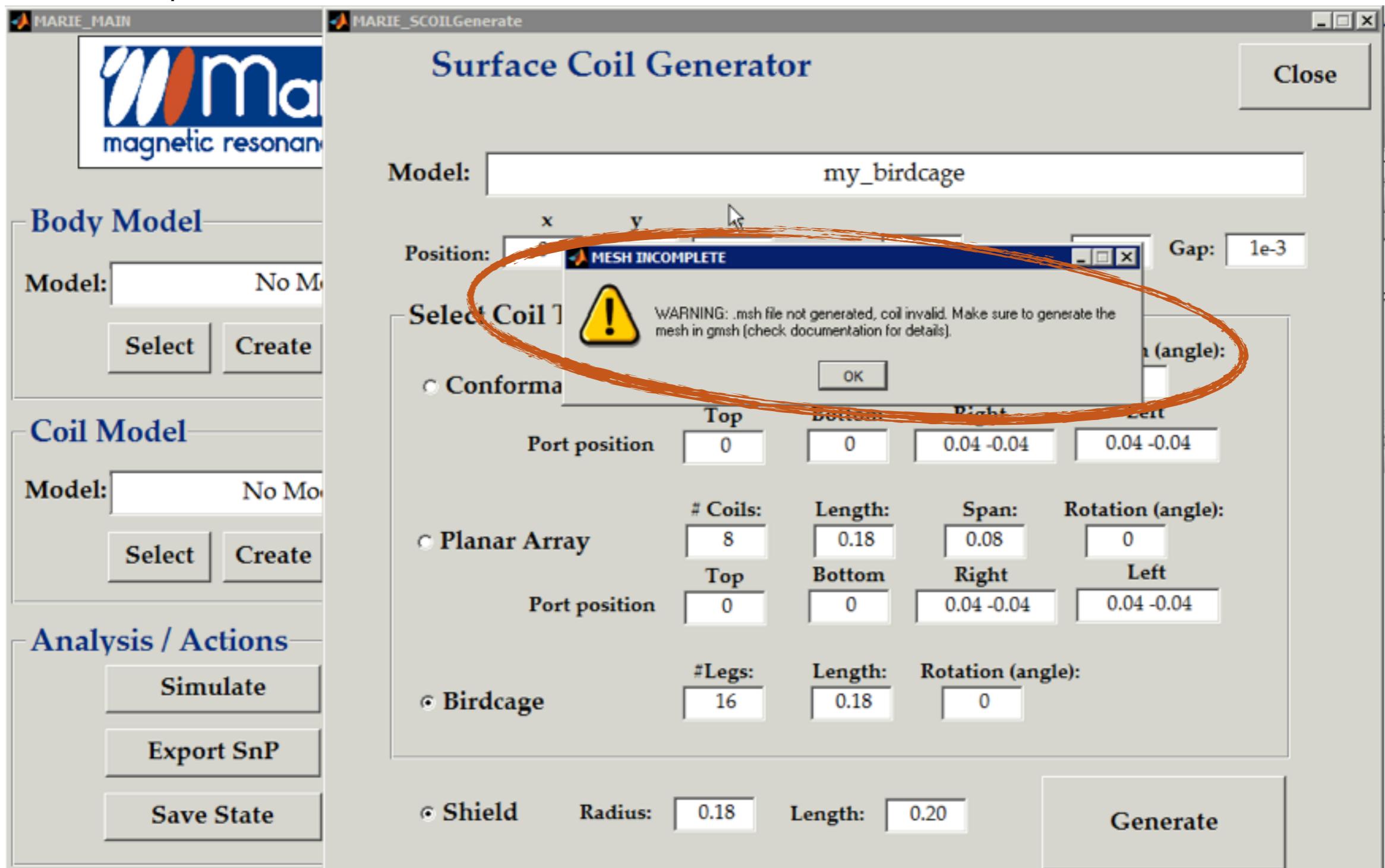
- ♦ Create a coil model

- ♦ if we do not push save, the .msh file will not be generated



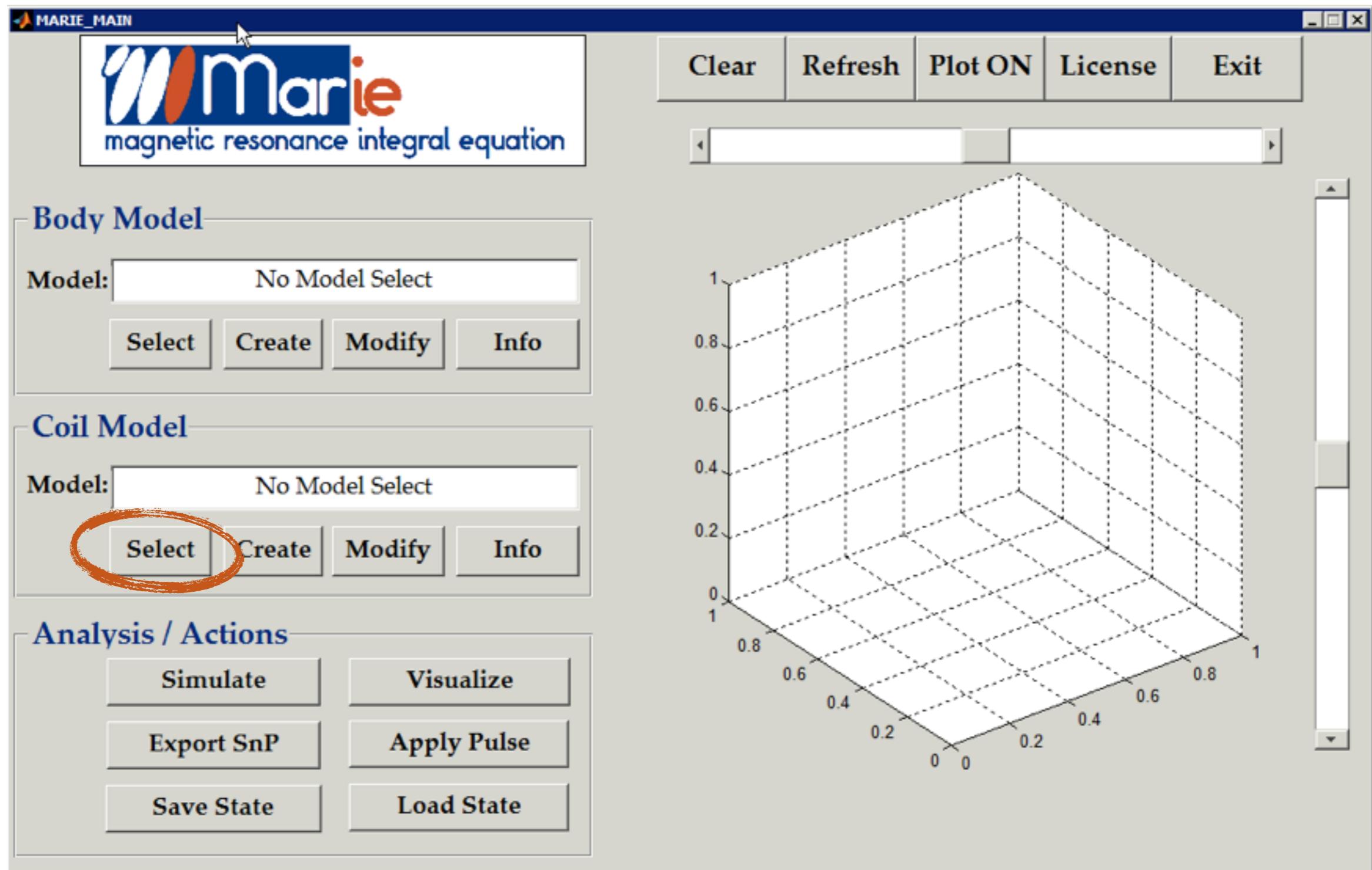
## ♦ Create a coil model

- ♦ and a warning will be issued about the invalid model
  - ♦ Always remember to save the mesh!!!

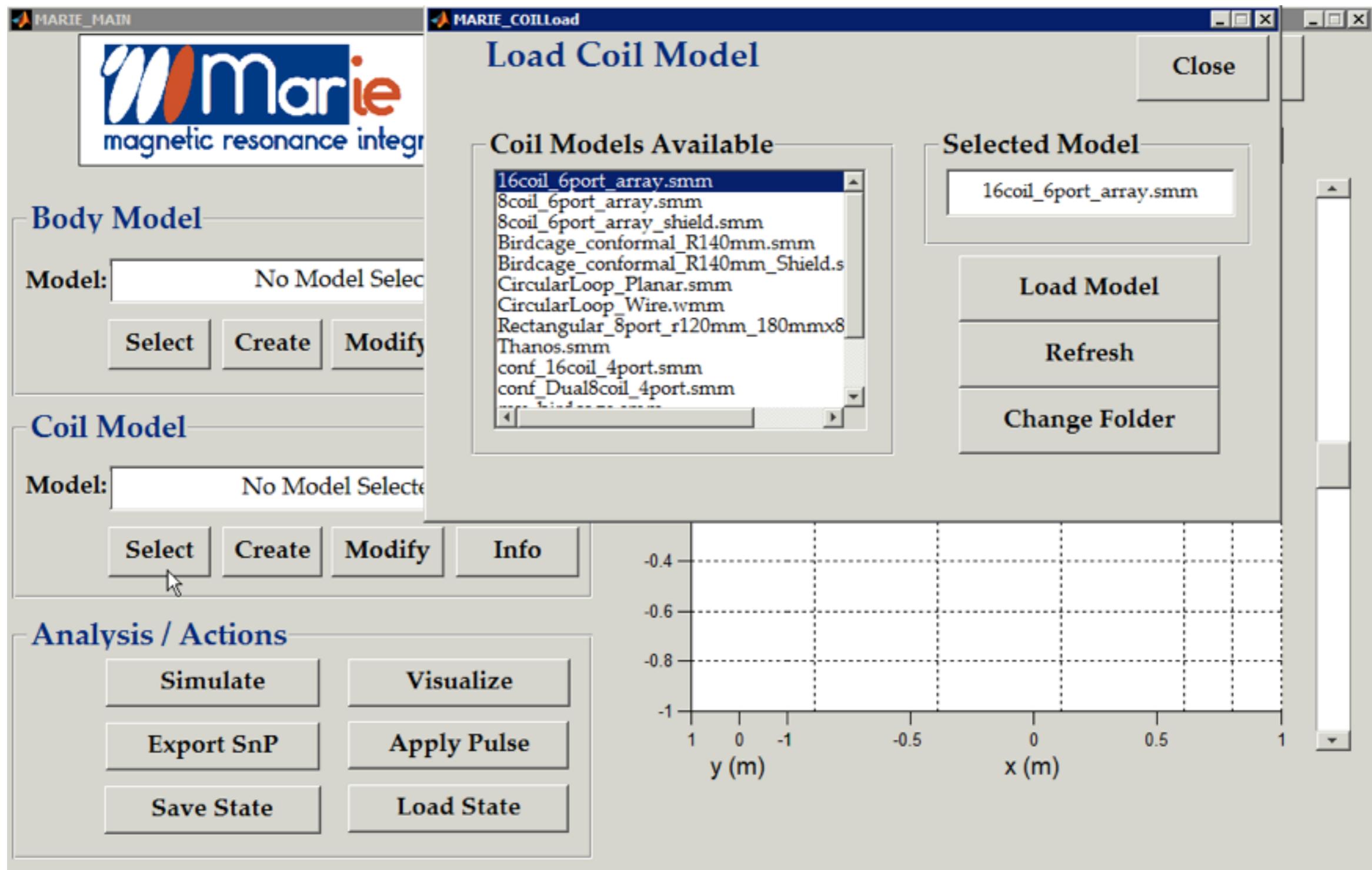


- ◆ Load coil model

- ◆ push “select” button to load a model

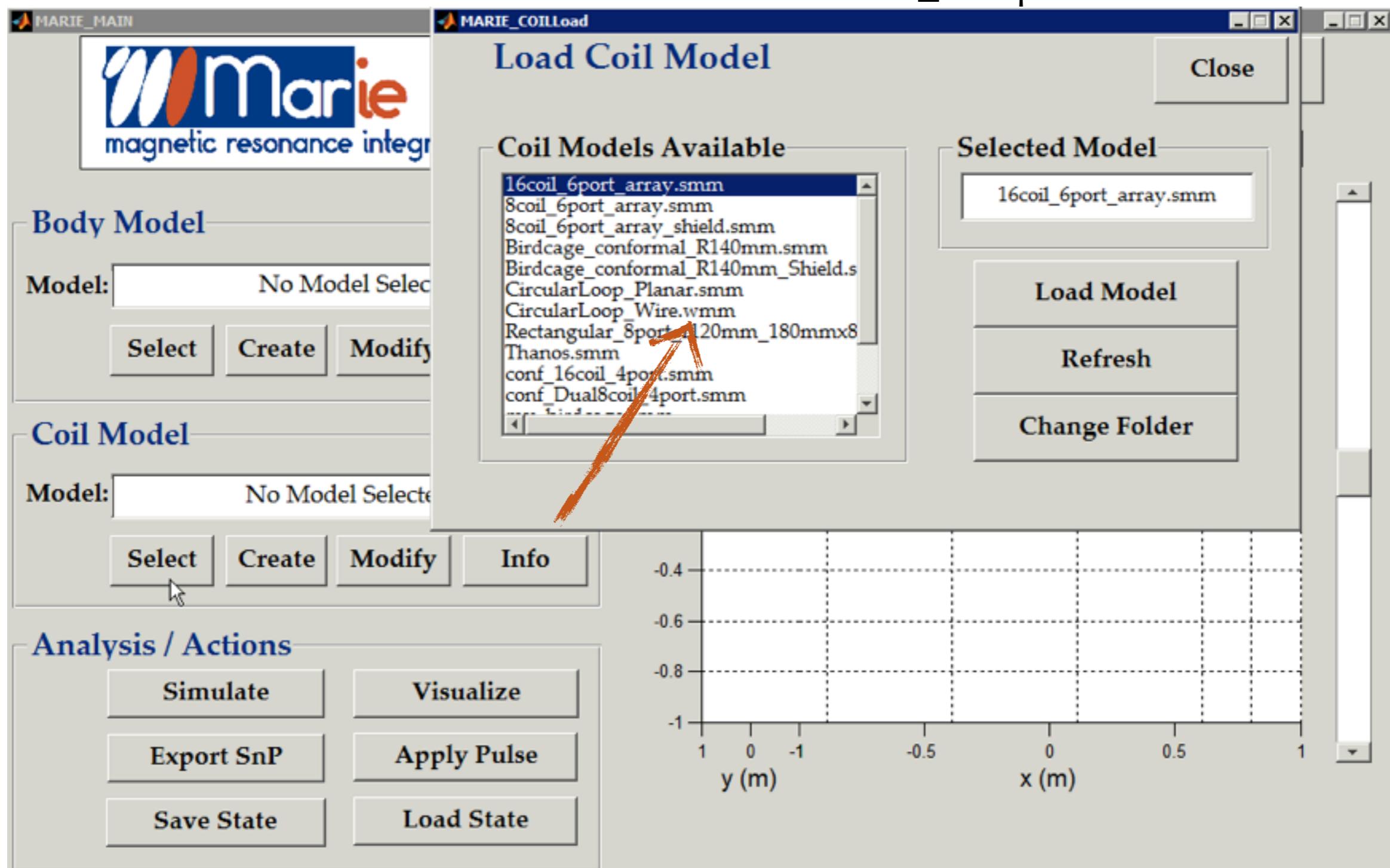


- ◆ Load coil model
  - ◆ dialogue to load coil model will open



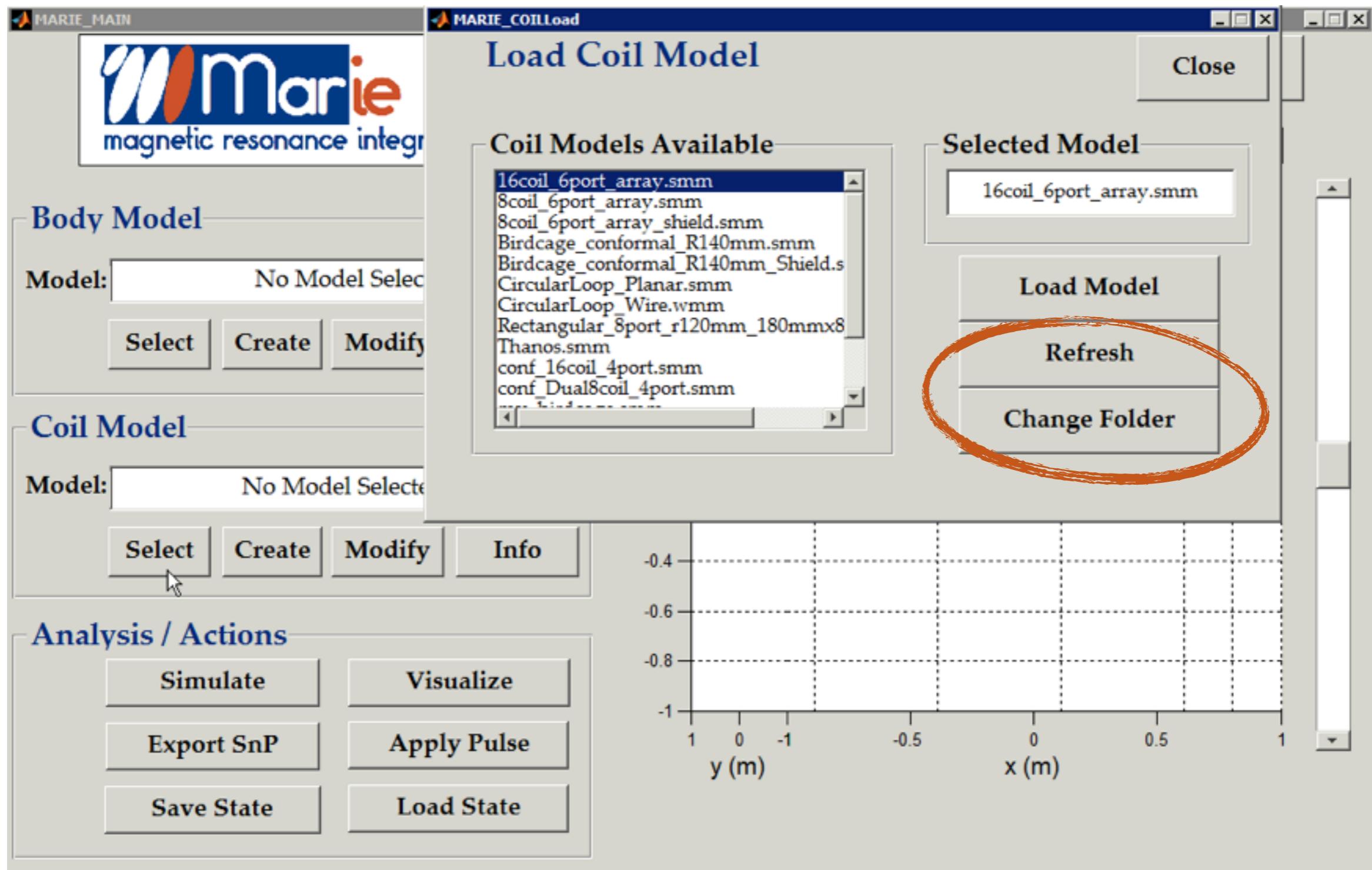
## ◆ Load coil model

- ◆ available models, surface (.smm) or wire (.wmm), are listed:
  - ◆ more details of the coil models data in MARIE\_coil.pdf



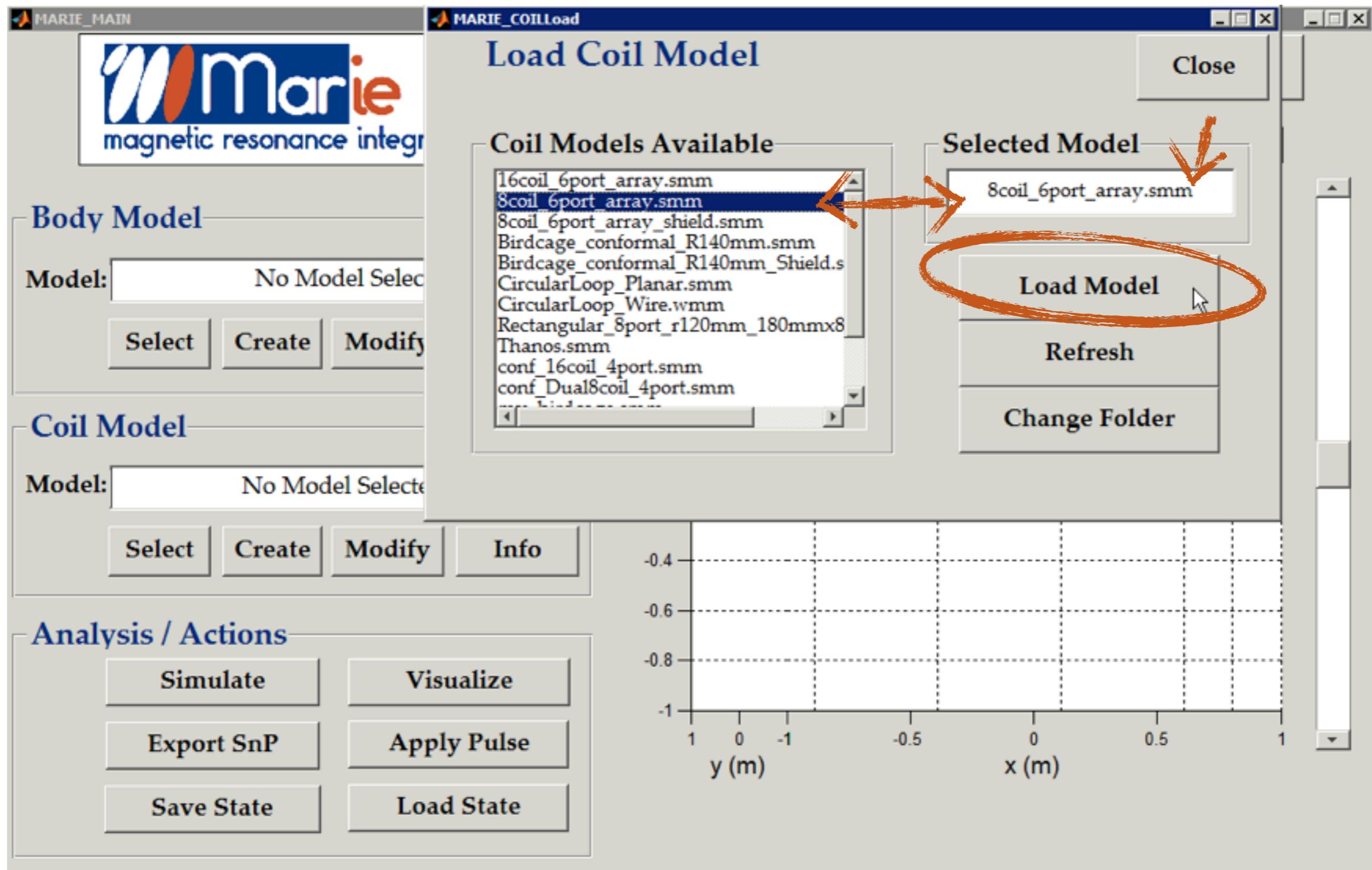
## ◆ Load coil model

- ◆ “change folder” and “refresh” can be used to find the desired model



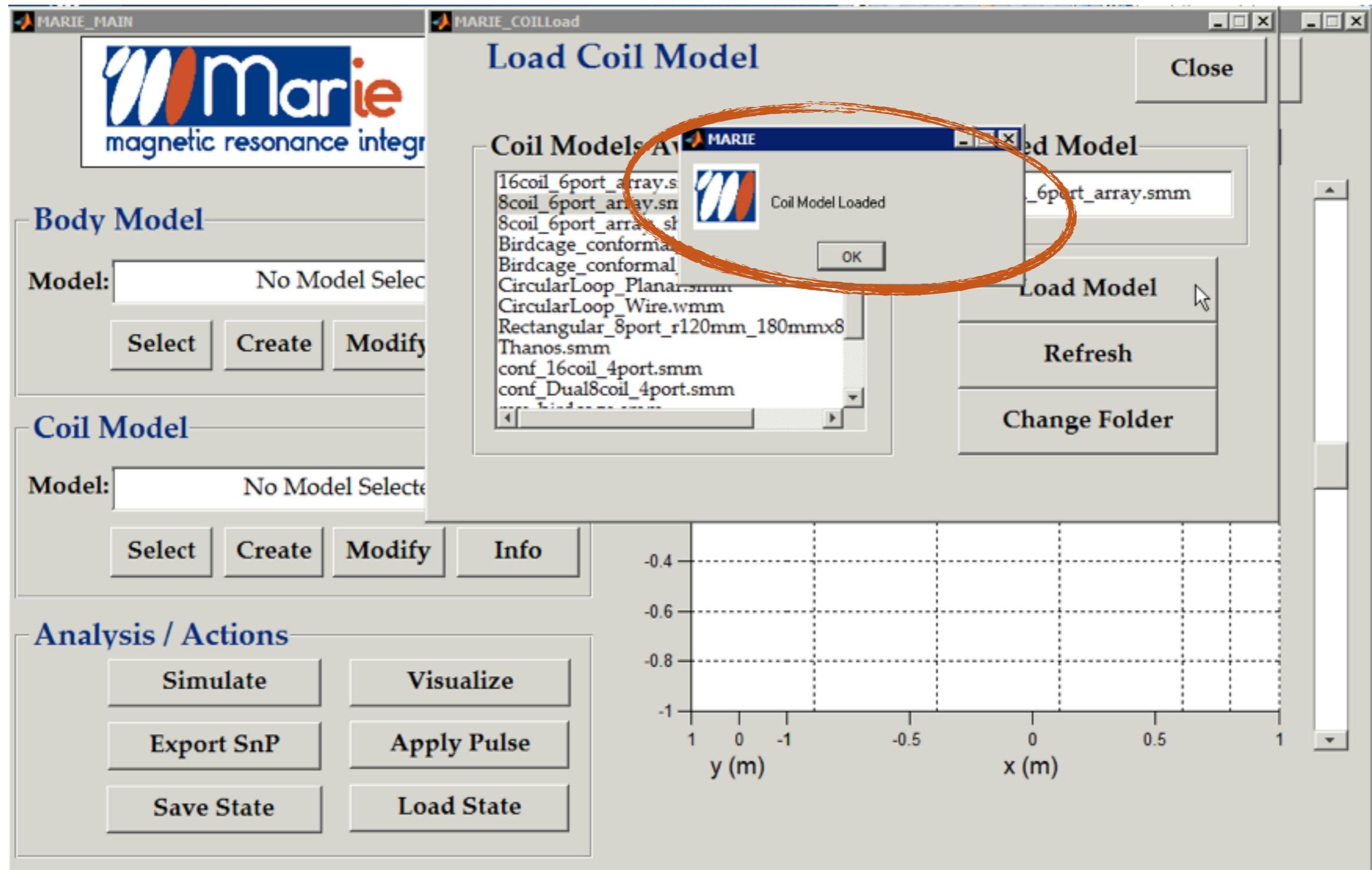
## ◆ Load coil model

- ◆ select the desired model and push “load model”



- ◆ Load coil model

- ◆ confirmation window will appear when loaded



## ◆ Load coil model

- ◆ info about the model will be printed in command line

**Command Window**

```

File: 8coil_6port_array.msh
Parse complete
  2000 nodes
  0 element points
  192 lines
  2560 triangles
  0 quadrangles

```

>> whos

| Name   | Size   | Bytes    | Class  | Attributes |
|--------|--------|----------|--------|------------|
| COIL   | 1x1    | 53072318 | struct | global     |
| FIGIDX | 1x1    | 8        | double | global     |
| PULSE  | 1x1    | 1760     | struct | global     |
| RHBM   | 1x1    | 1090     | struct | global     |
| SOL    | 1x1    | 1584     | struct | global     |
| p      | 1x6341 | 12682    | char   |            |

**MARIE\_MAIN**

**Body Model**

Model: No Model Select

Select Create Modify Info

**Coil Model**

Model: Array of 8 6-port planar coils

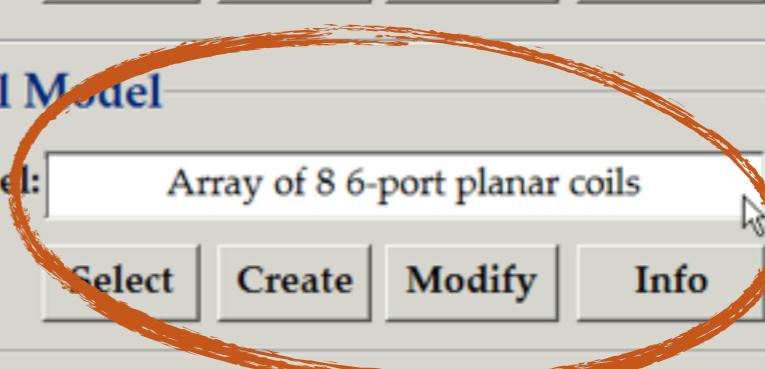
Select Create Modify Info

**Analysis / Actions**

Simulate Visualize

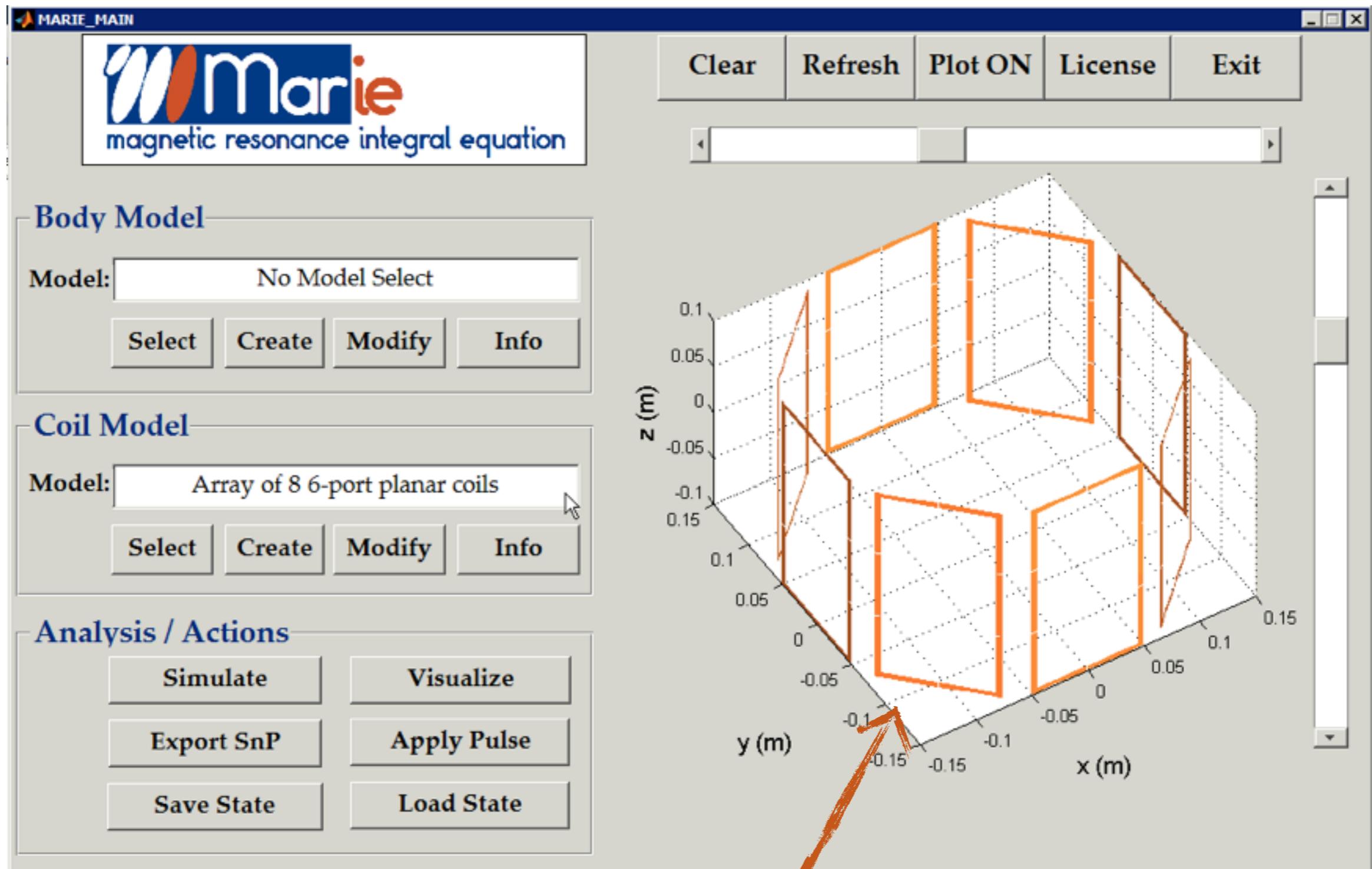
Export SnP Apply Pulse

Save State Load State

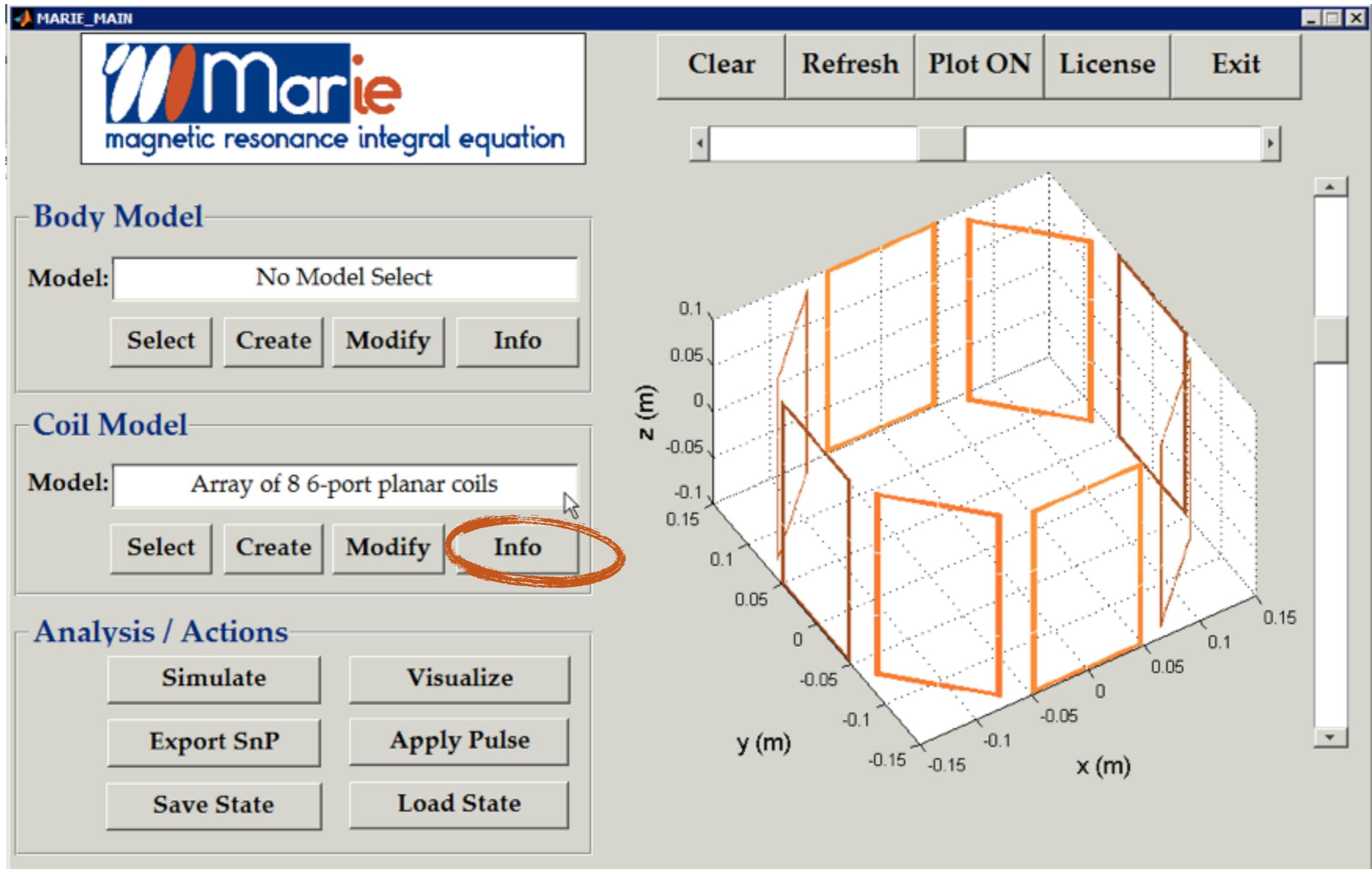
- ◆ Load coil model

- ◆ model will be plot in figure



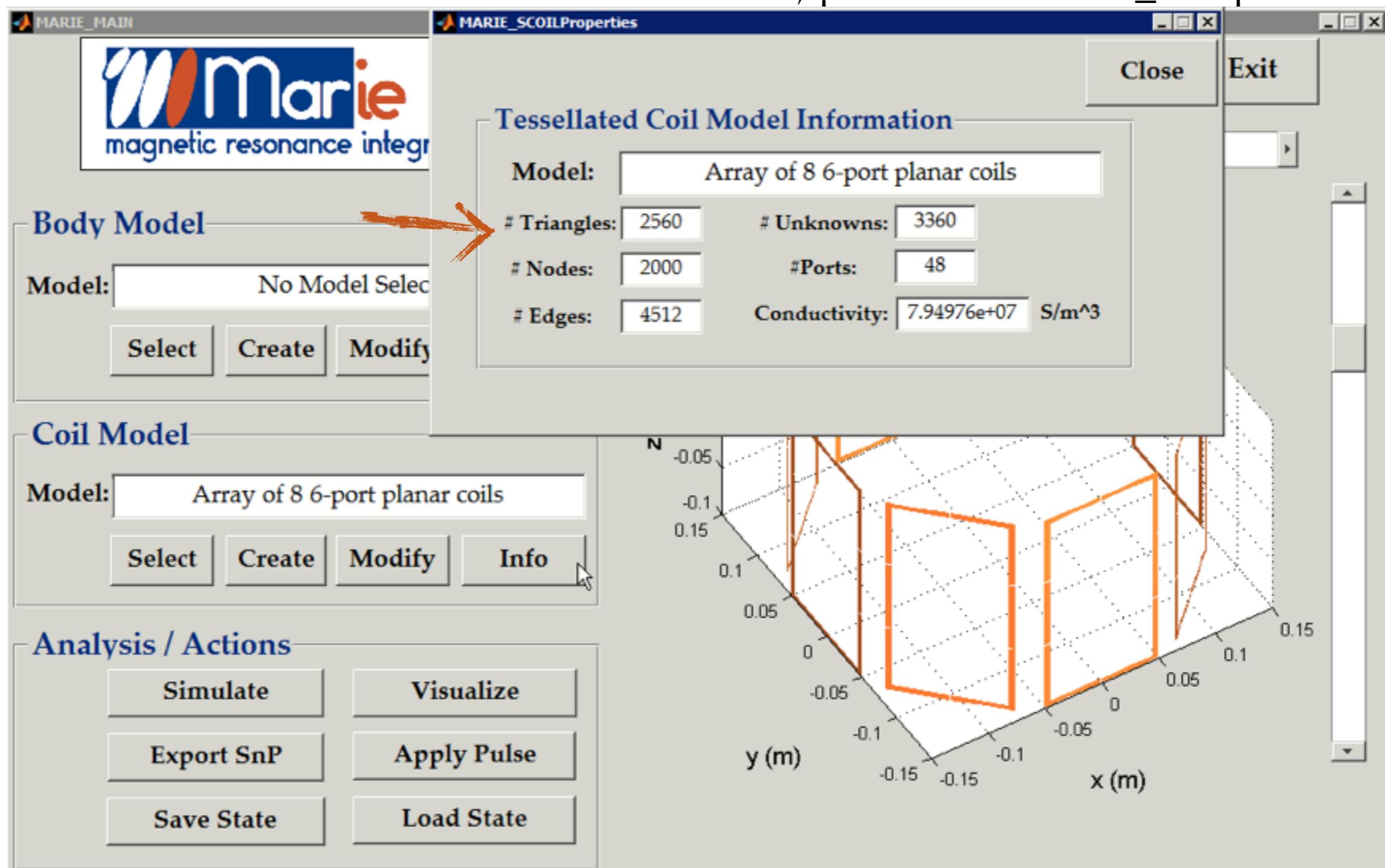
- ◆ View coil model information

- ◆ information about the coil model can be seen by pushing “info”



## ♦ View coil model information

- ♦ in this case is a surface coil model (from a .smm file)
  - ♦ for more details of the coil models data, please see MARIE\_coil.pdf



## ♦ View coil model information

- ♦ in this case is a surface coil model (from a .smm file)
- ♦ detail of the structure data related to surface (type 'S') models

**Command Window**

```

COIL      1x1          53072318  struct
FIGIDX   1x1           8    double
PULSE     1x1          1760  struct
RHBM      1x1          1090  struct
SOL       1x1          1584  struct
p         1x6341        12682 char

>> COIL

COIL =

```

name: 'Array of 8 6-port planar coils'  
type: 'S'  
Rhocoil: 1.257900000000000e-08  
Thickness: 5.000000000000000e-04  
index: [4512x1 double]  
etod: [3x2560 double]  
node: [3x2000 double]  
edge: [2x4512 double]  
elem: [4x2560 double]  
index\_elem: [1x1 struct]  
Ct: [3x2560 double]  
Ln: [3x2560 double]  
Pn: [3x3x2560 double]  
port: [1x48 struct]  
Pcoil: []  
Ncoil: []  
Dwire: []

↓

**MARIE\_MAIN**

**Body Model**

Model: No Model Selected

Select Create Modify

**MARIE\_SCOILProperties**

**Tessellated Coil Model Information**

Model: Array of 8 6-port planar coils

# Triangles: 2560 # Unknowns: 3360  
# Nodes: 2000 # Ports: 48  
# Edges: 4512 Conductivity: 7.94976e+07 S/m

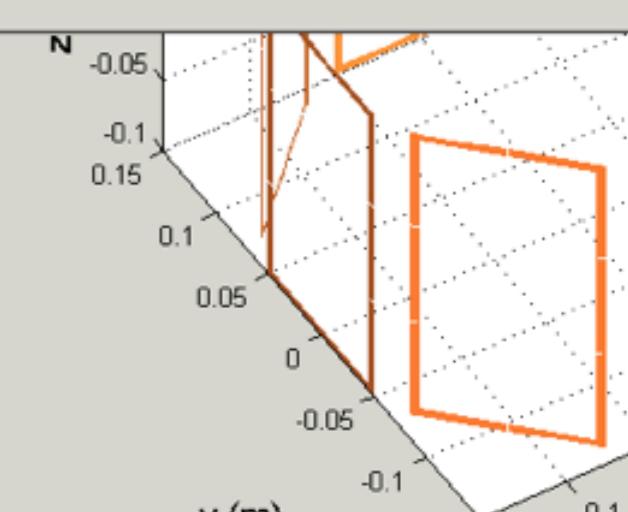
**Coil Model**

Model: Array of 8 6-port planar coils

Select Create Modify Info

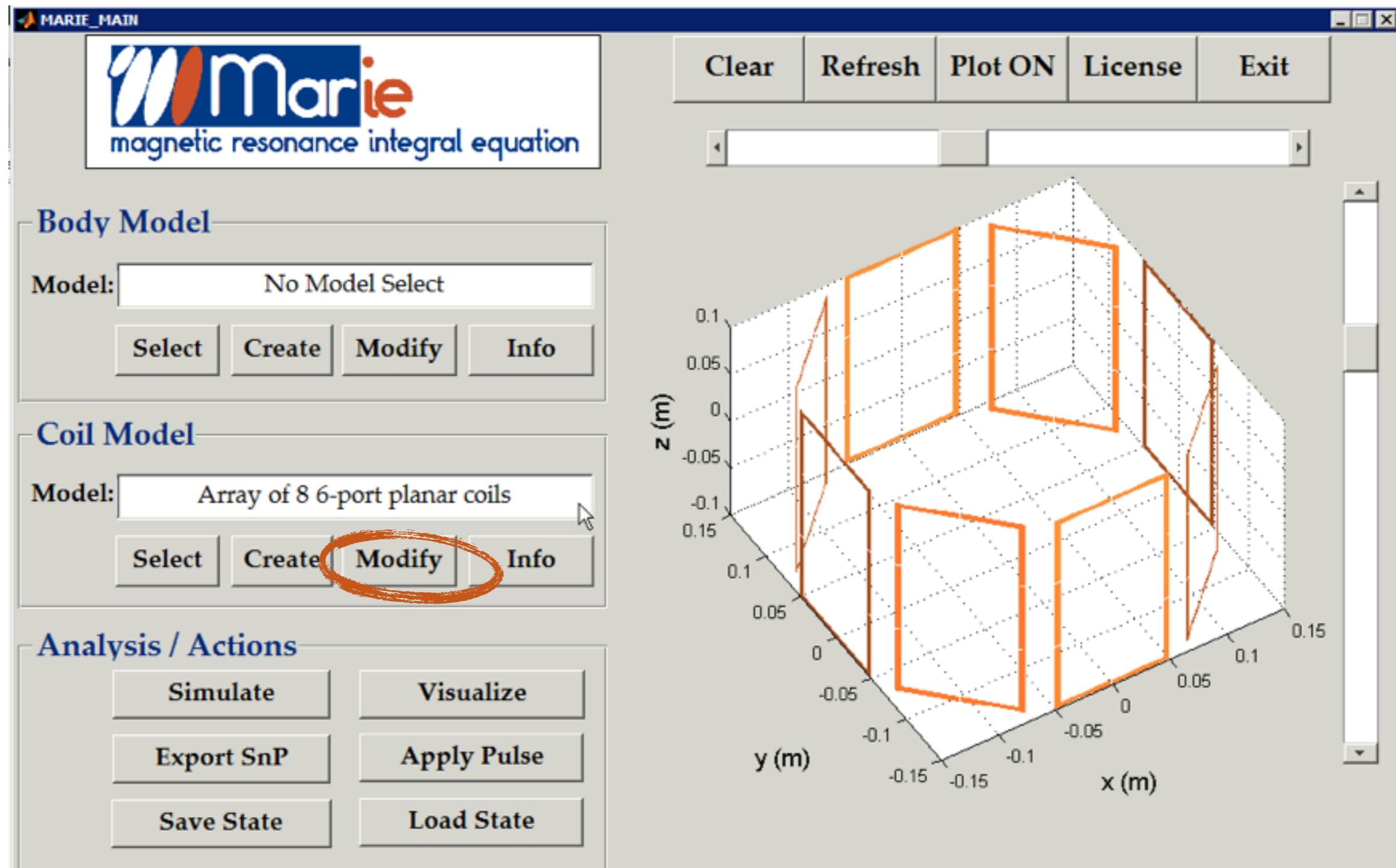
**Analysis / Actions**

Simulate Visualize  
Export SnP Apply Pulse  
Save State Load State

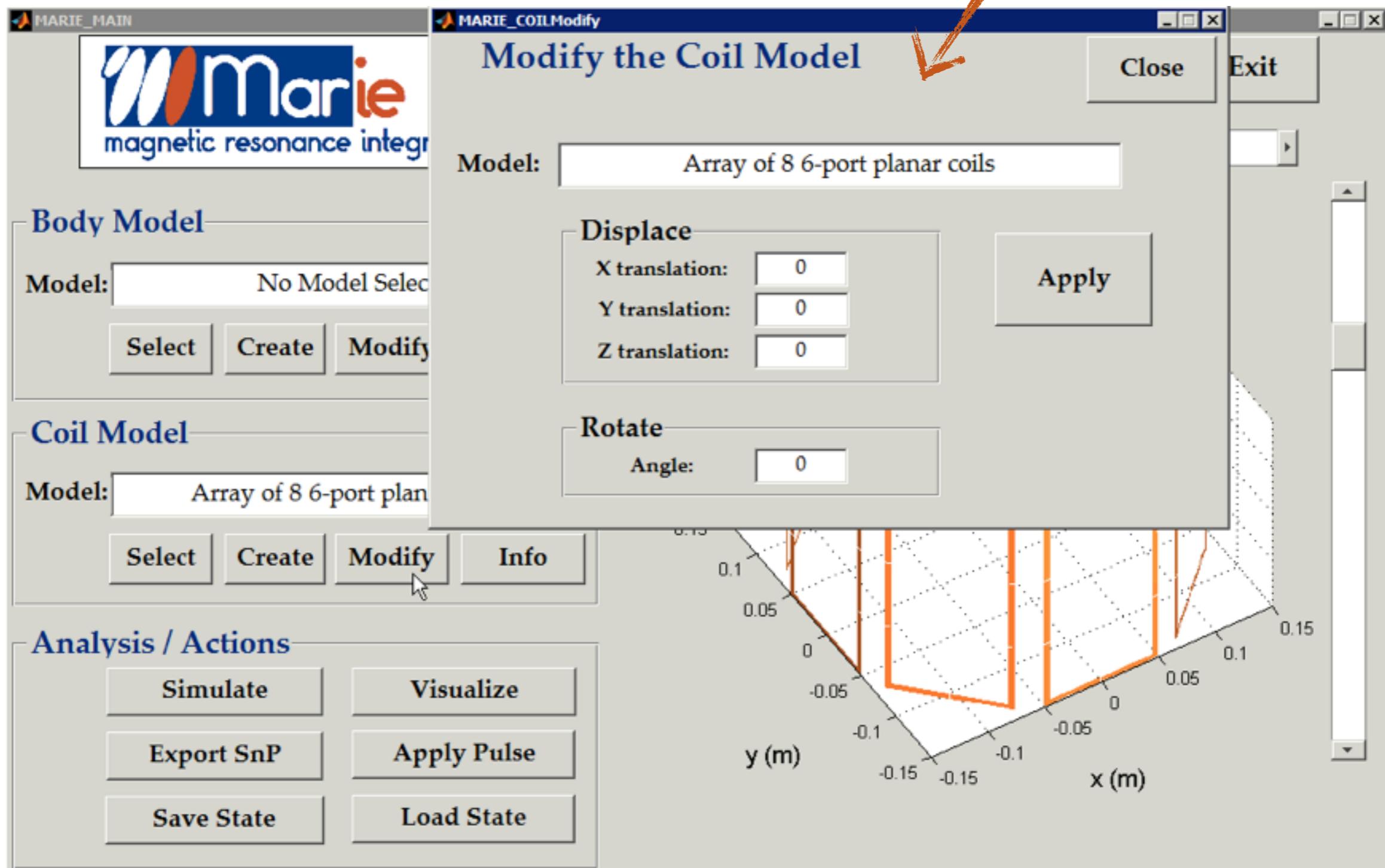


- ♦ **Modify the coil model**

- ♦ push “modify” to change the position and or rotation of the coil

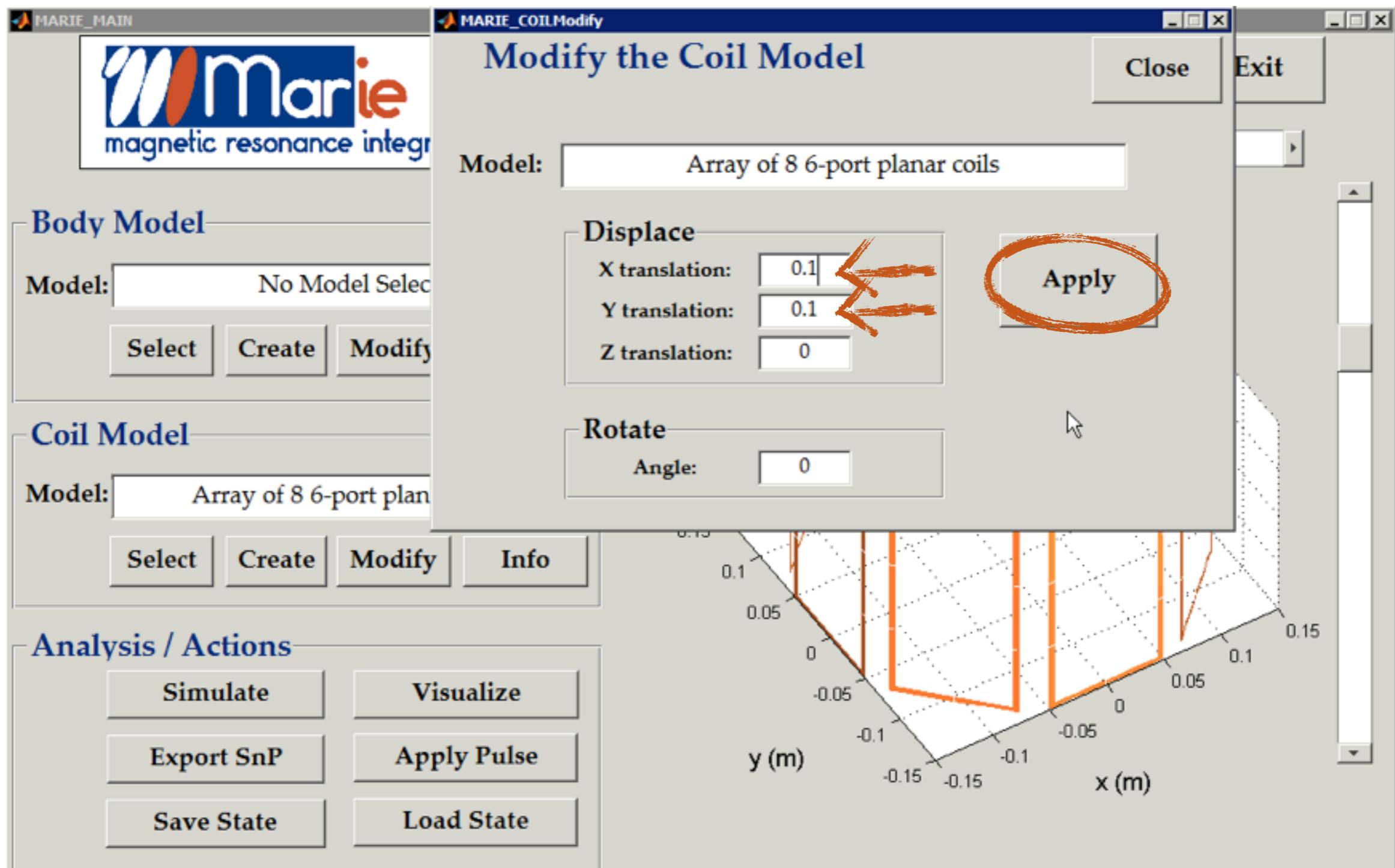


- ♦ Modify the coil model
  - ♦ dialogue will appear



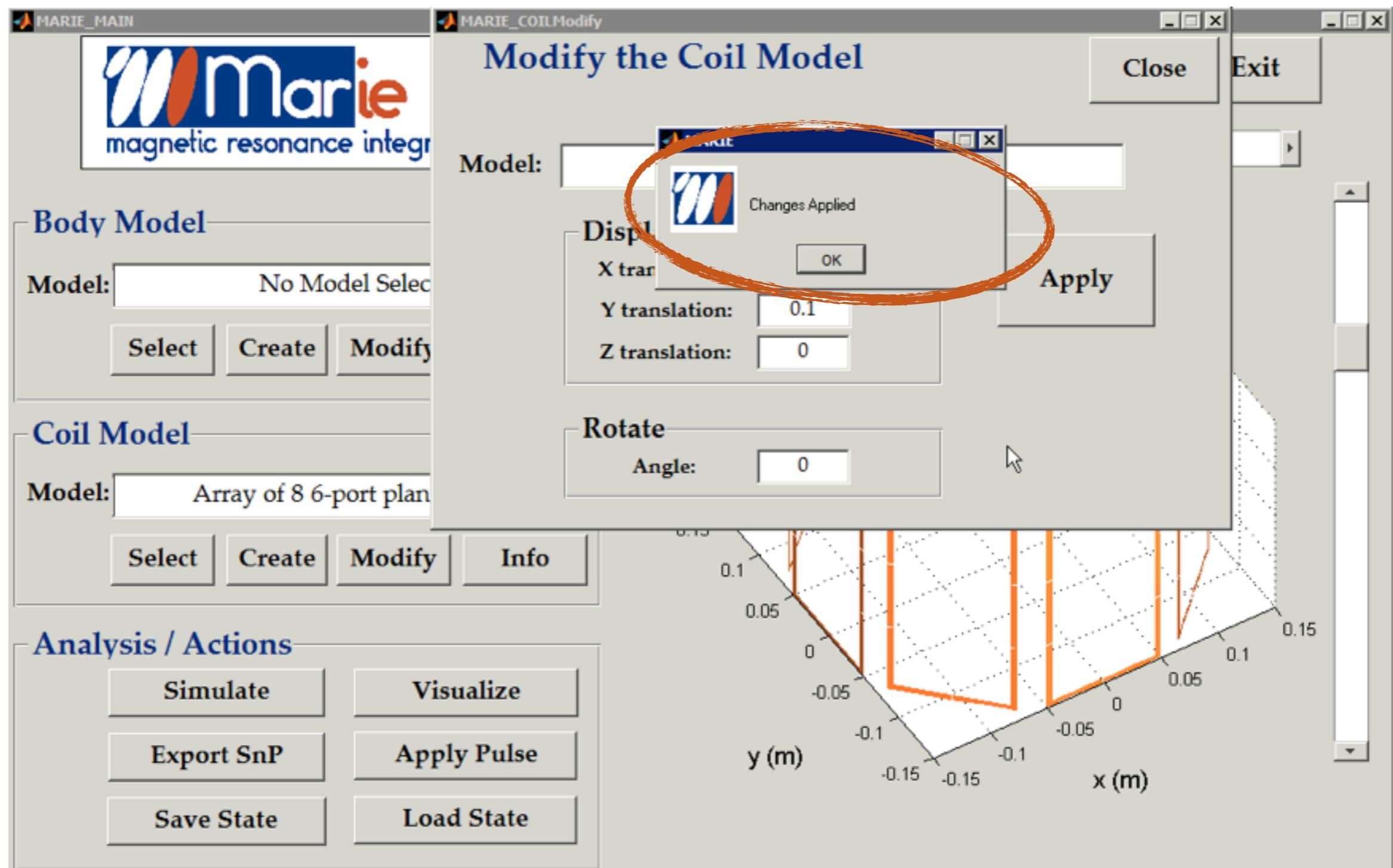
## ♦ Modify the coil model

- ♦ introduce the changes and press apply button

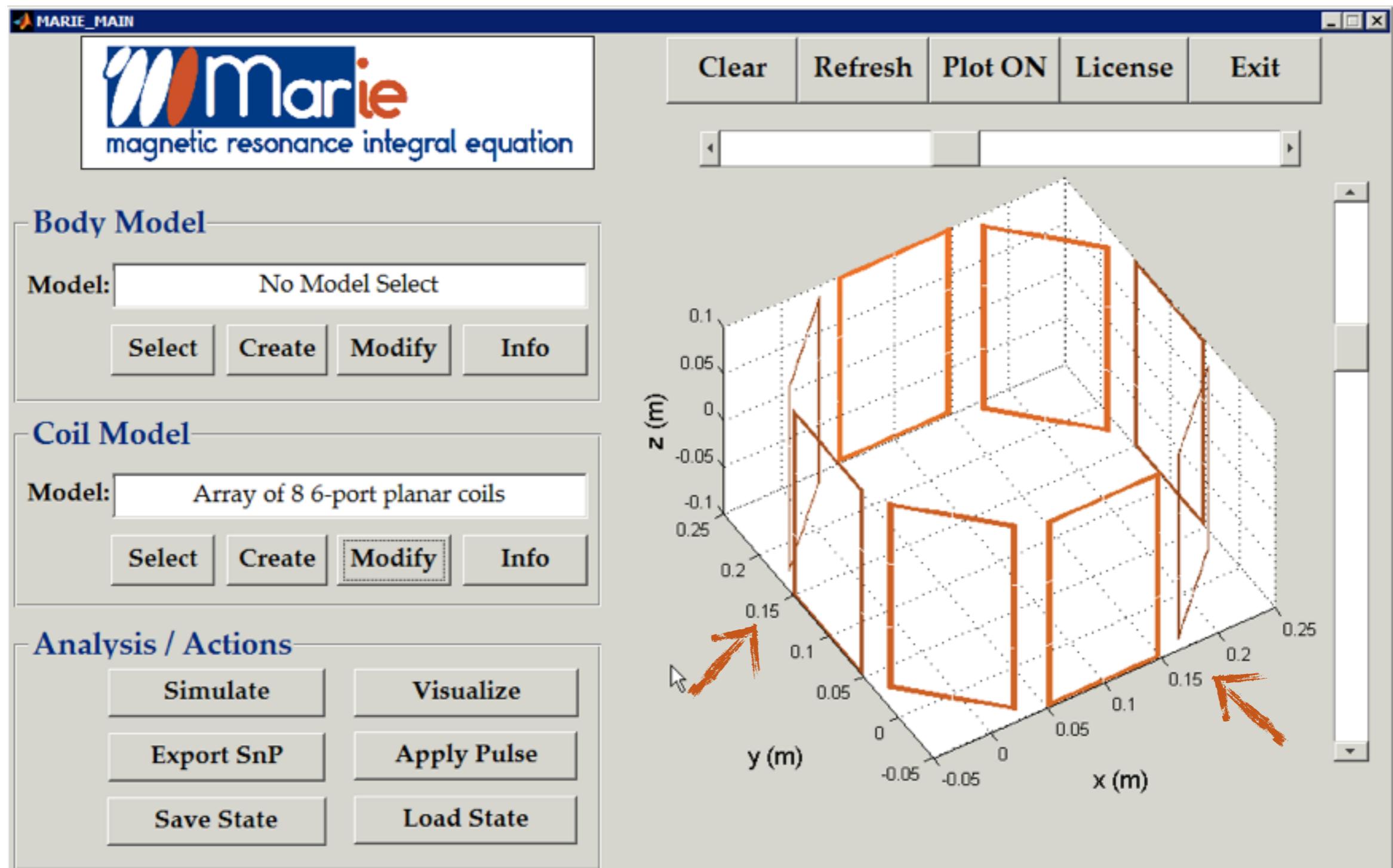


- ♦ Modify the coil model

- ♦ confirmation window when changes have been applied

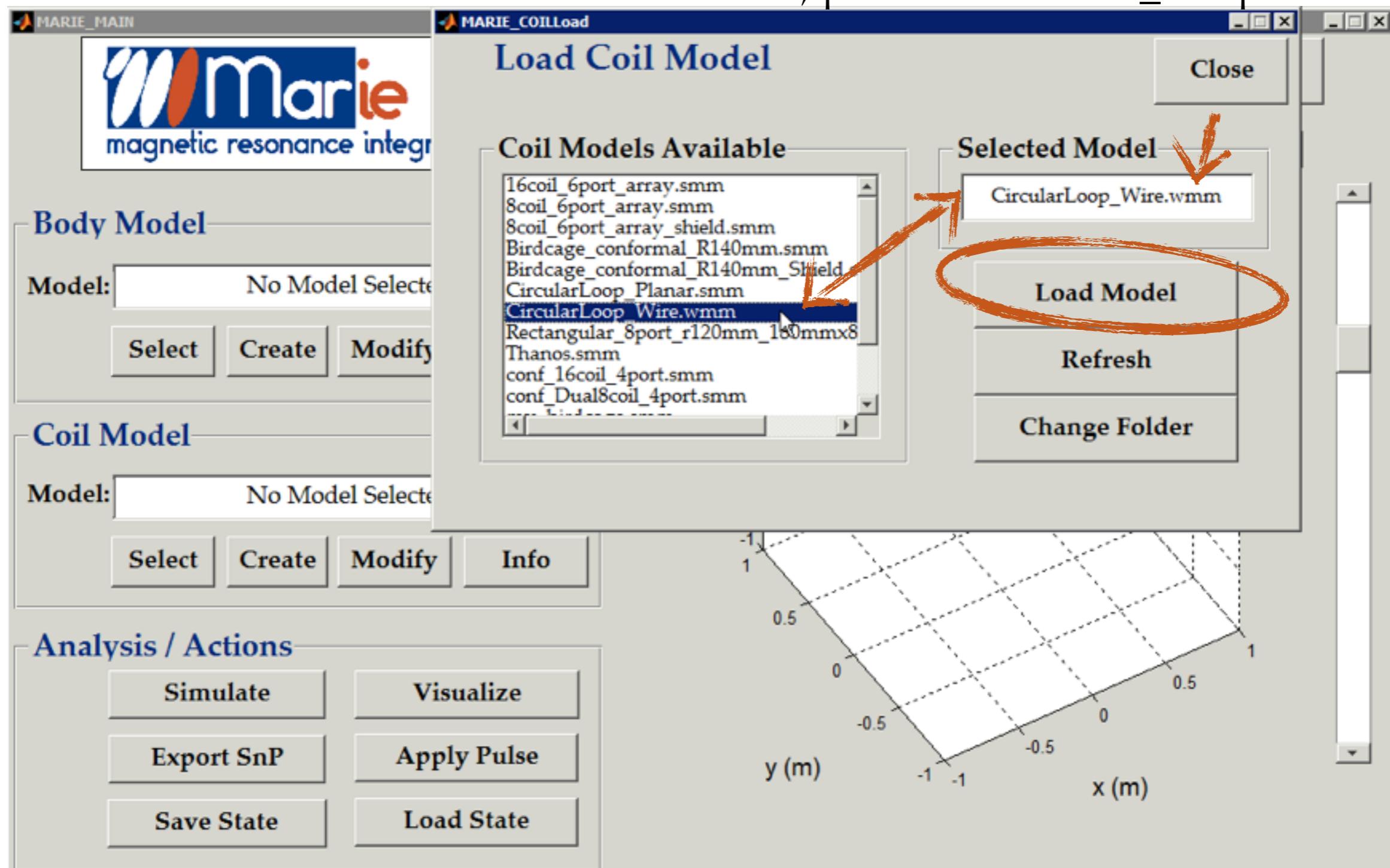


- ♦ Modify the coil model
  - ♦ changes can be seen on the figure limits

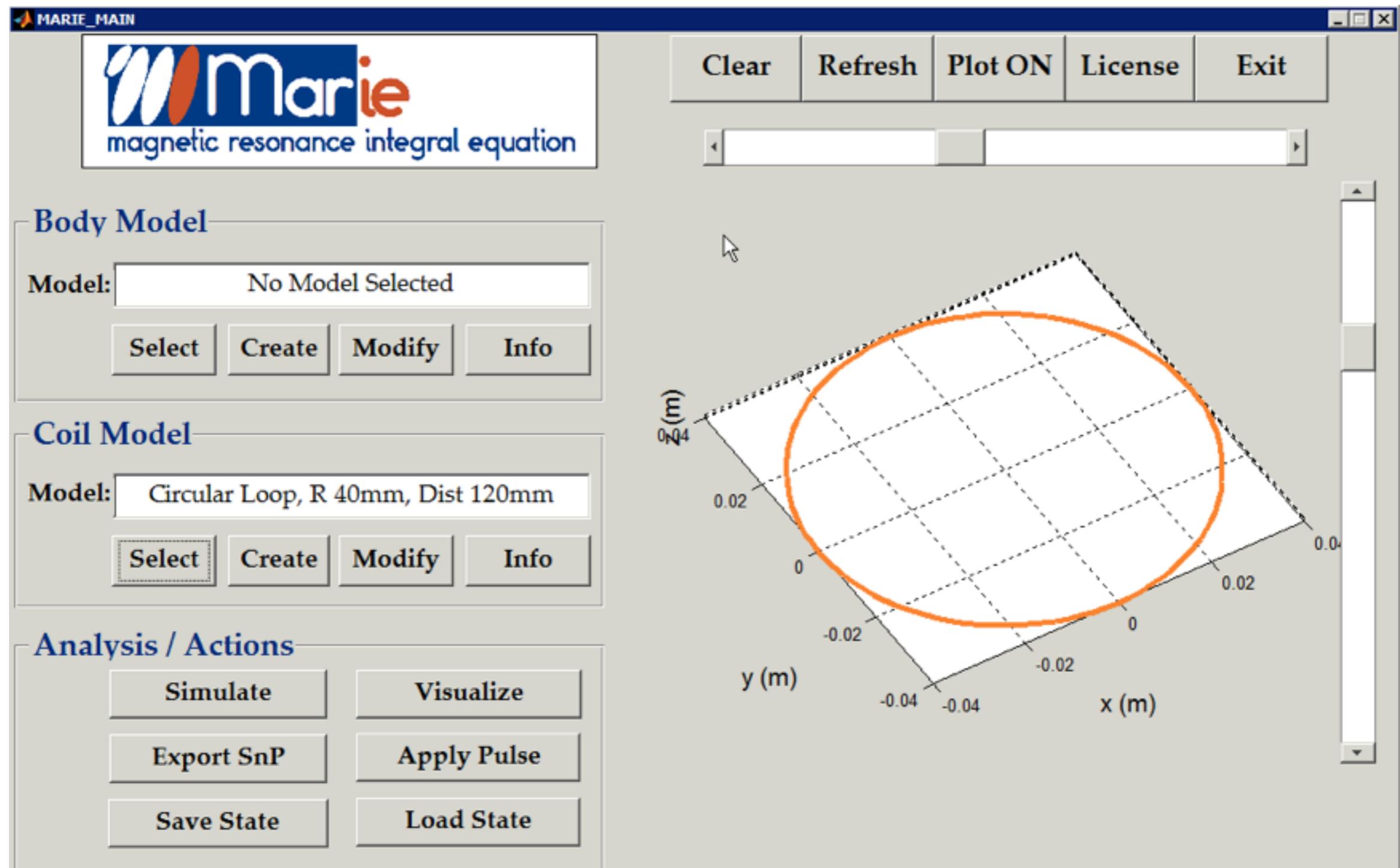


## ◆ Load coil model (wire model case)

- ◆ a wire coil model (.wmm) is loaded the same way
  - ◆ for more details of the coil models data, please see MARIE\_coil.pdf

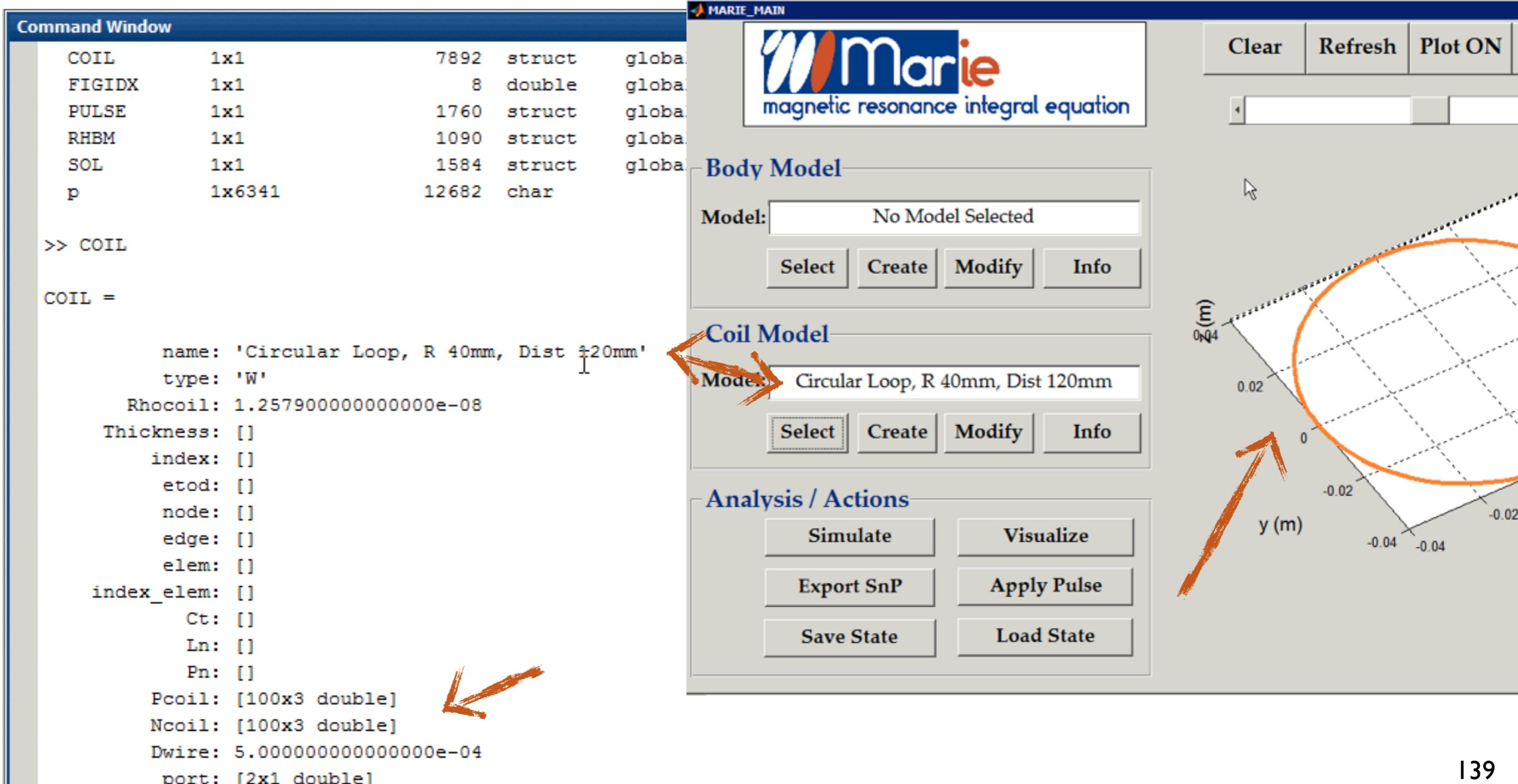


- ◆ Load coil model (wire model case)
  - ◆ coil will be shown in the figure



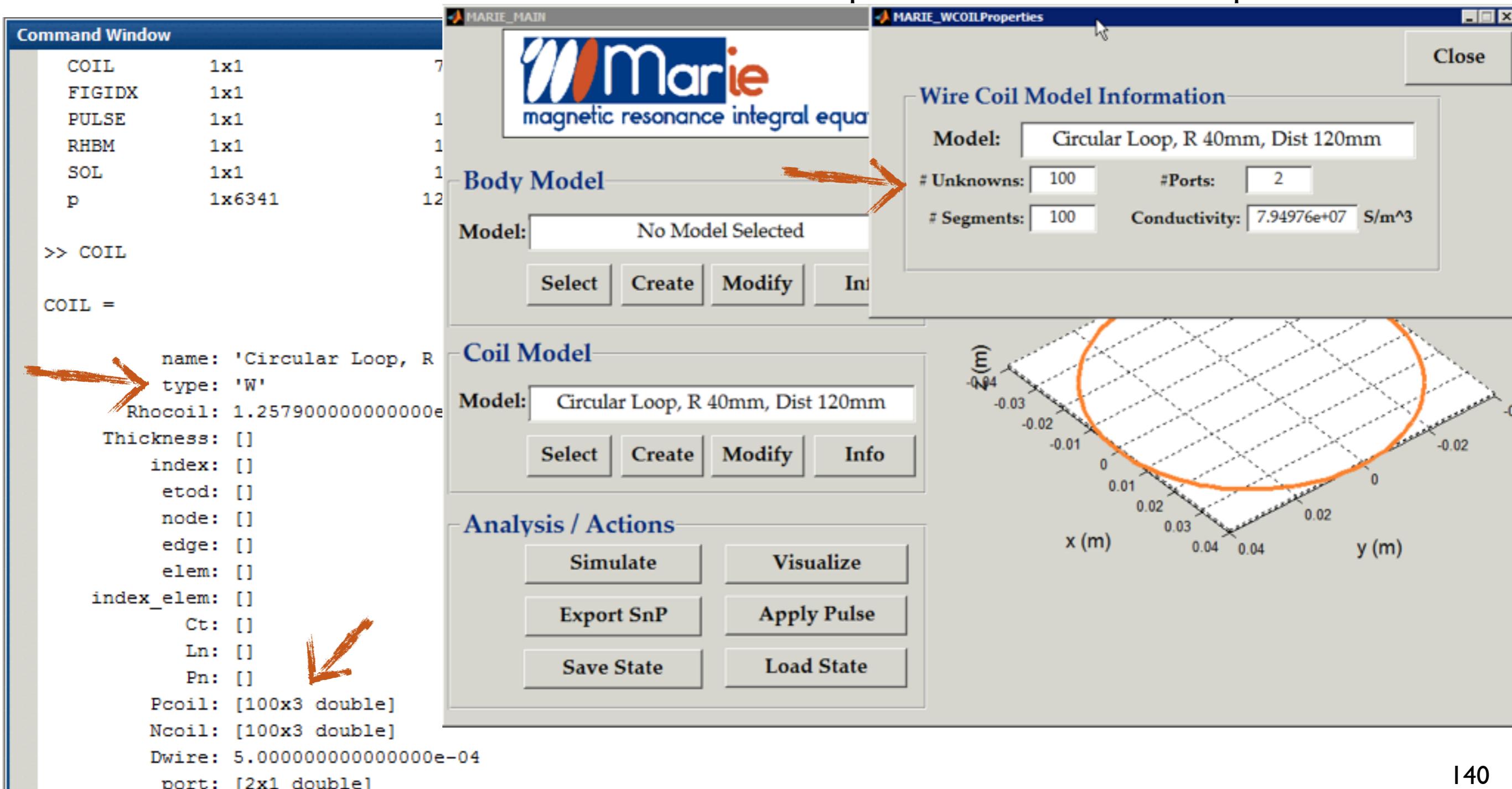
## ♦ Load coil model (wire model case)

- ♦ different fields of the COIL structure are used in this case
  - ♦ for more details of the coil models data, please see MARIE\_coil.pdf



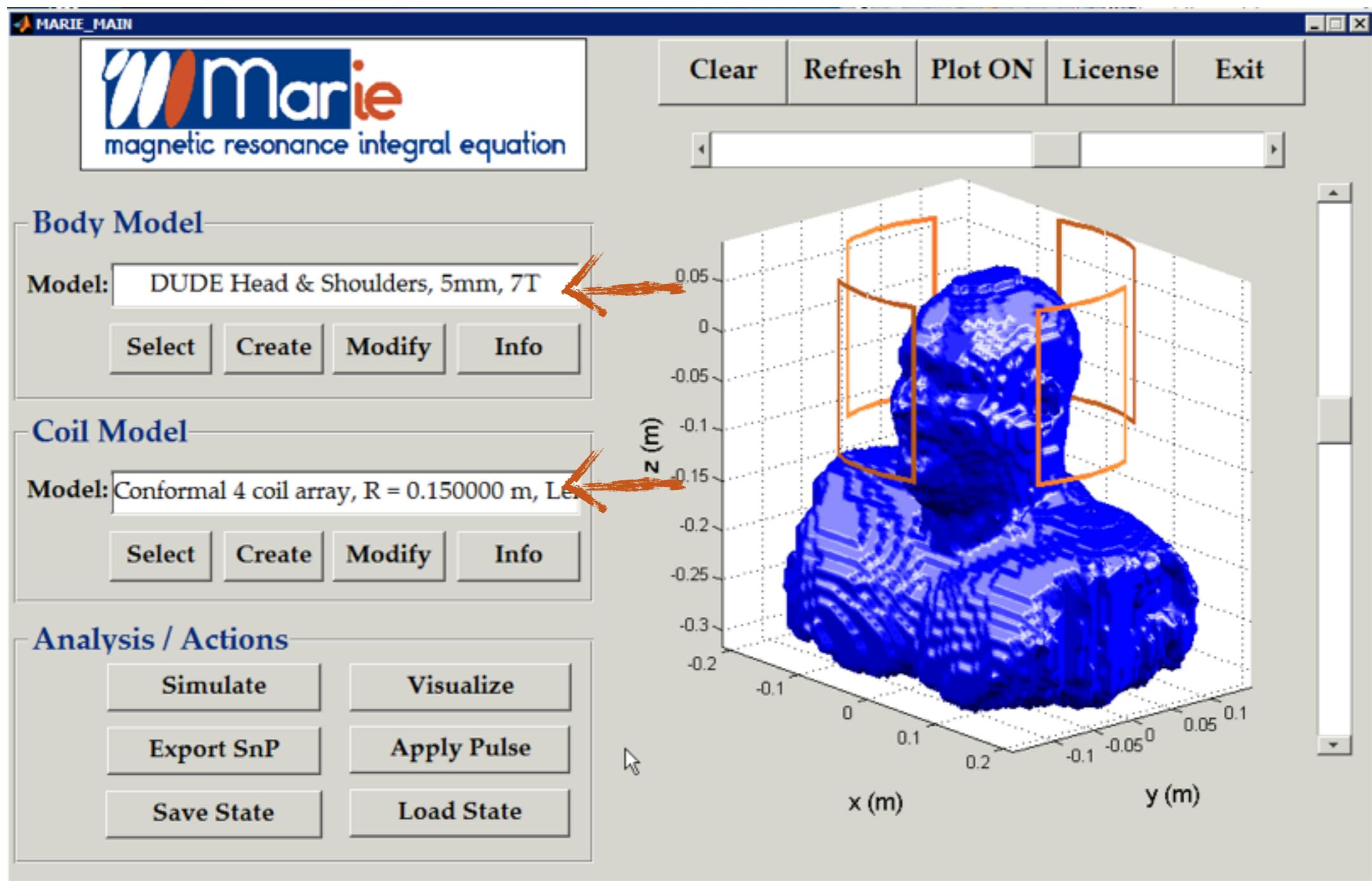
## ♦ View coil model information (wire model)

- ♦ and the info button shows wire model (type 'W') related information
  - ♦ for more details of the coil models data, please see MARIE\_coil.pdf

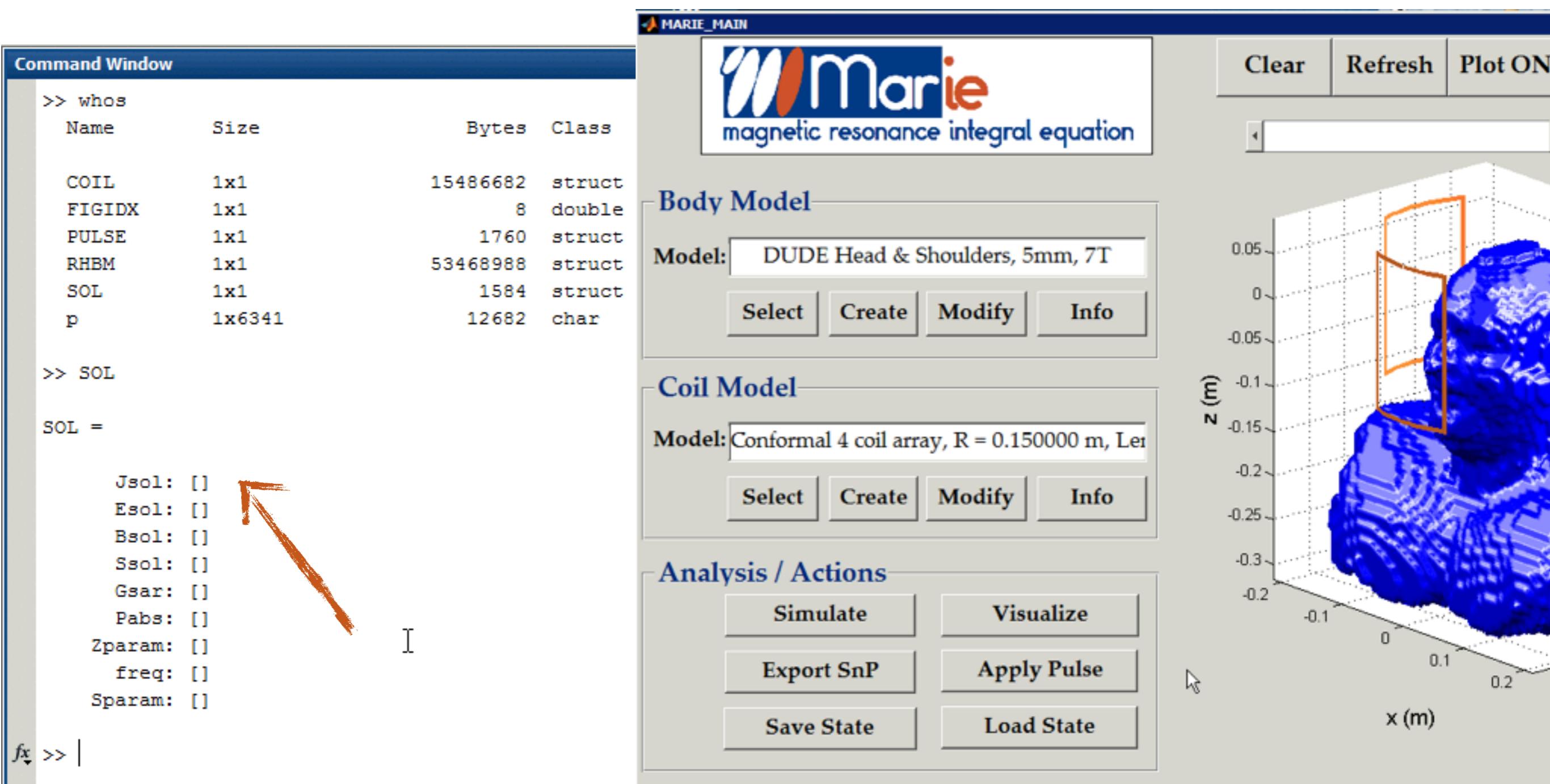


gui - analysis/actions box

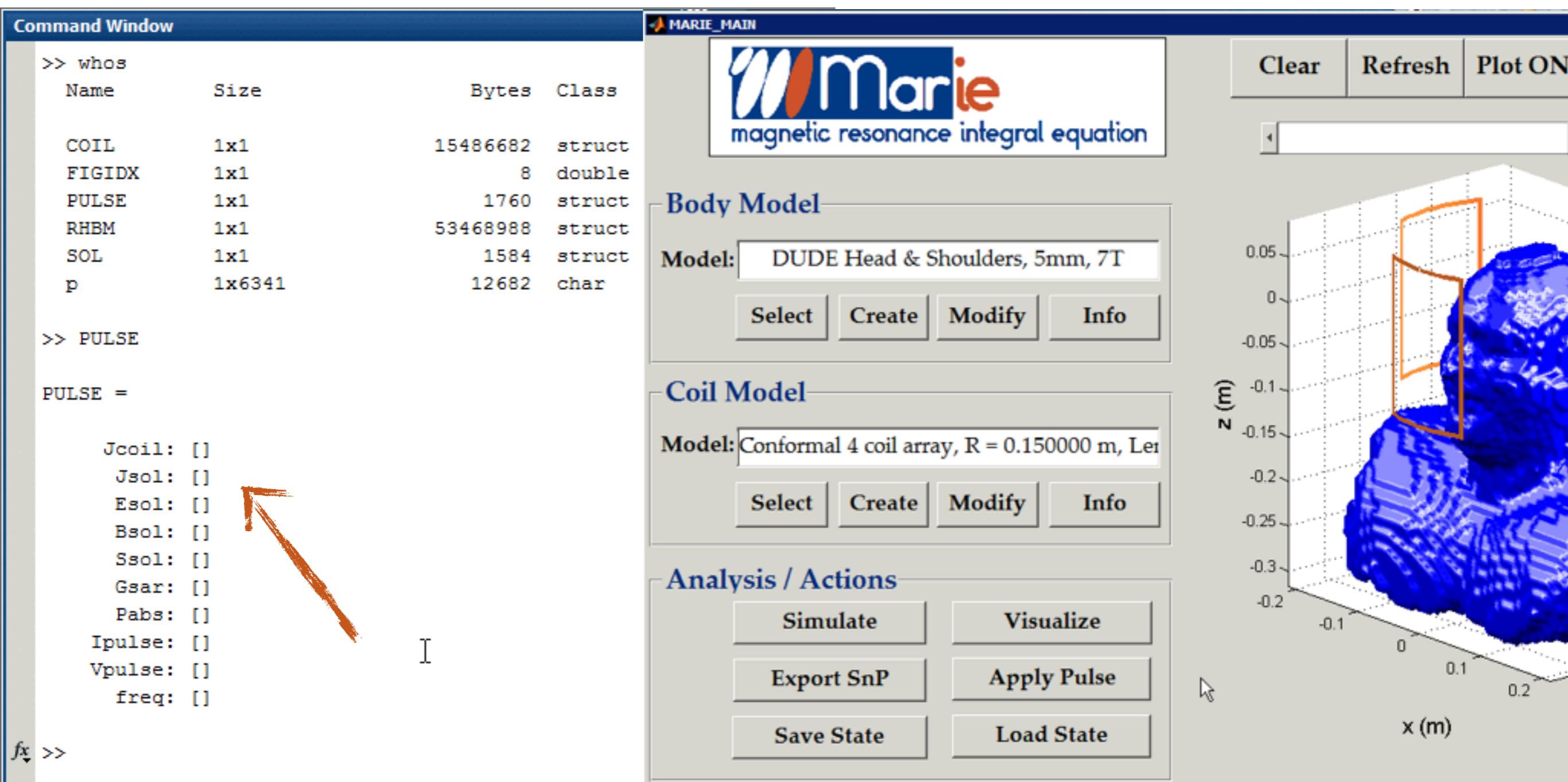
- ◆ Function box related to simulation, analysis and visualization
  - ◆ load the body and/or coil models



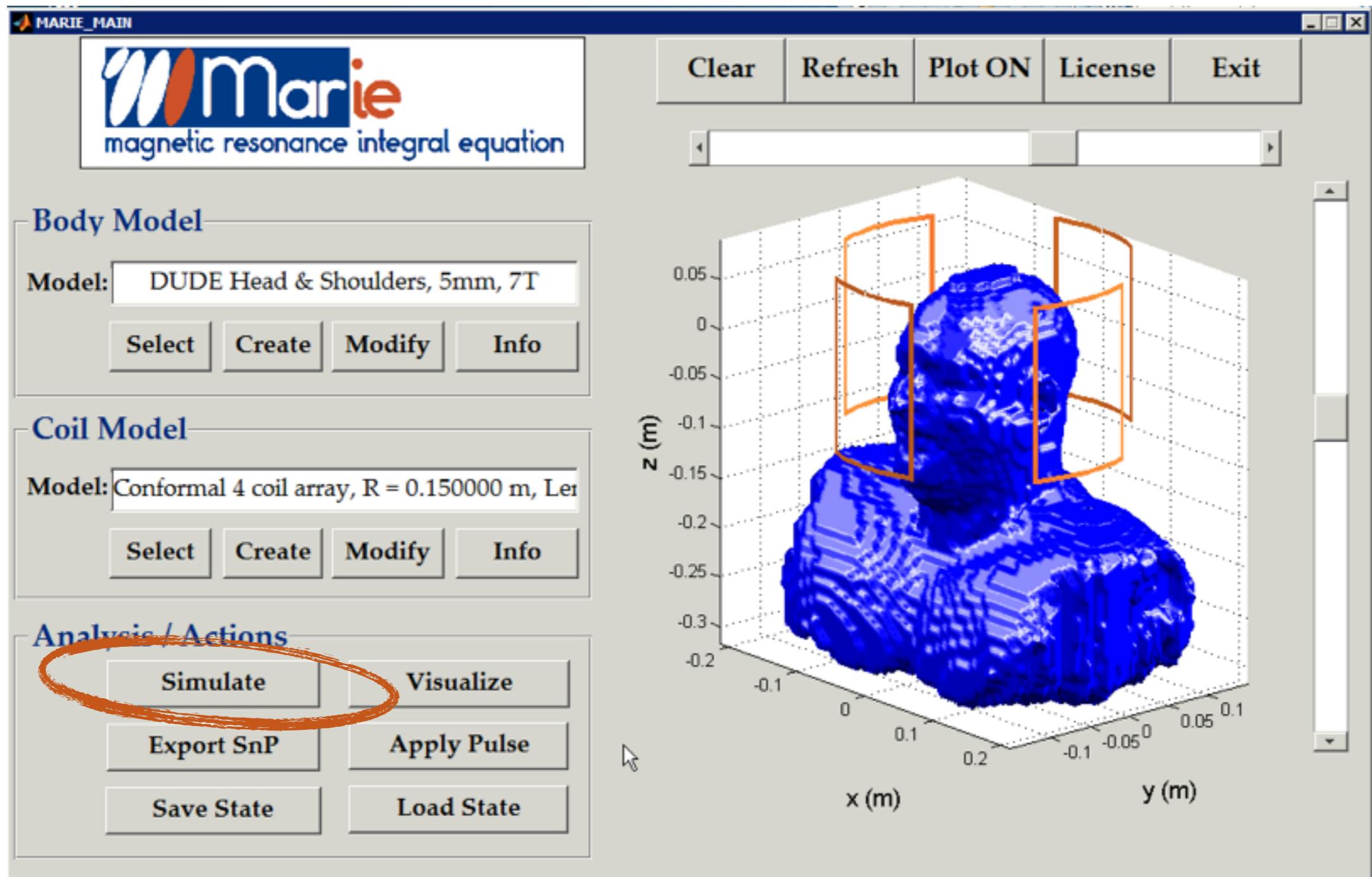
- ◆ Function box related to simulation, analysis and visualization
  - ◆ load the body and/or coil models
  - ◆ initially, the SOL structure is empty



- ◆ Function box related to simulation, analysis and visualization
  - ◆ load the body and/or coil models
  - ◆ and so is the PULSE structure

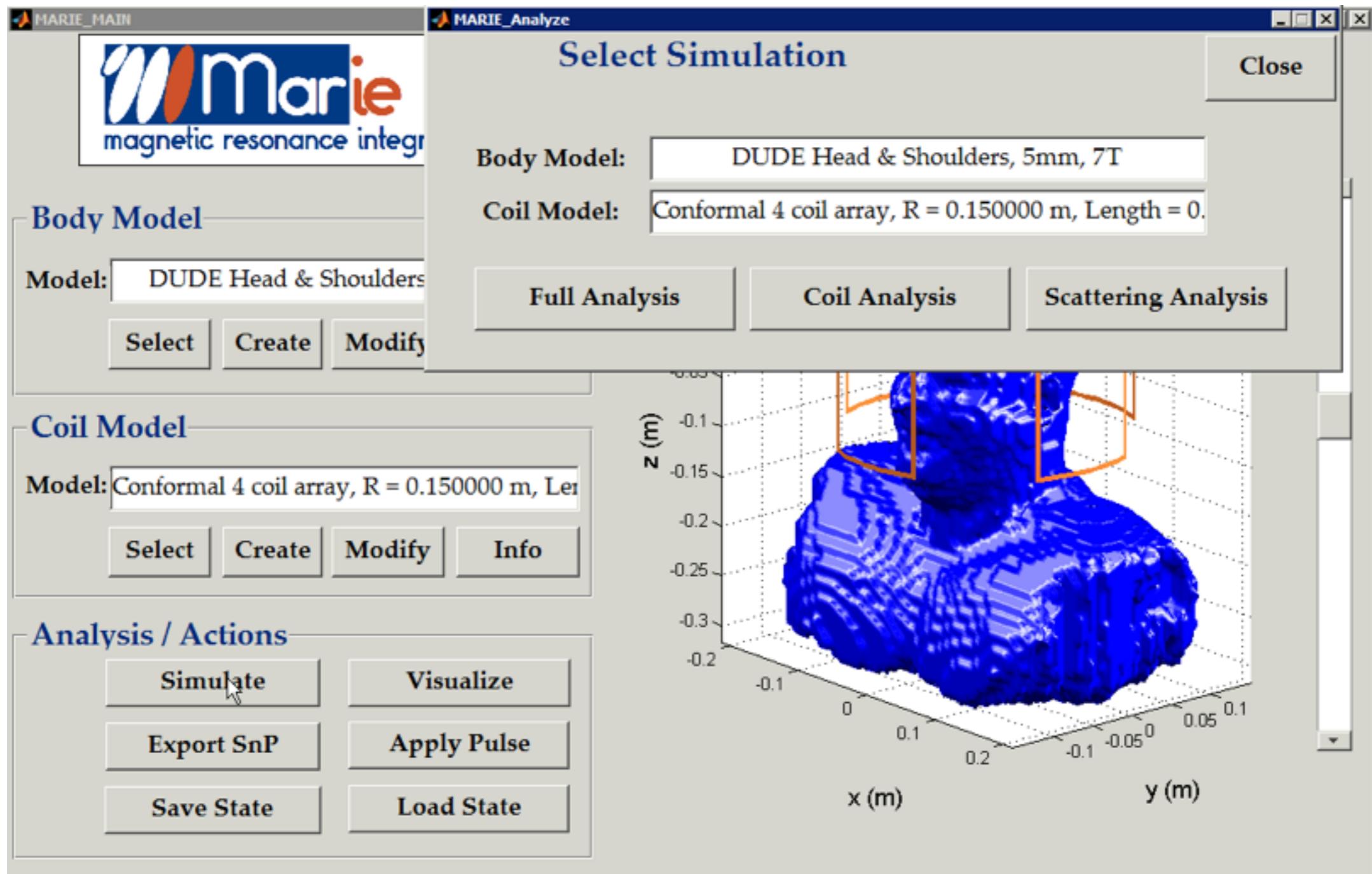


- ♦ Call the simulation engine
  - ♦ push “simulate” button

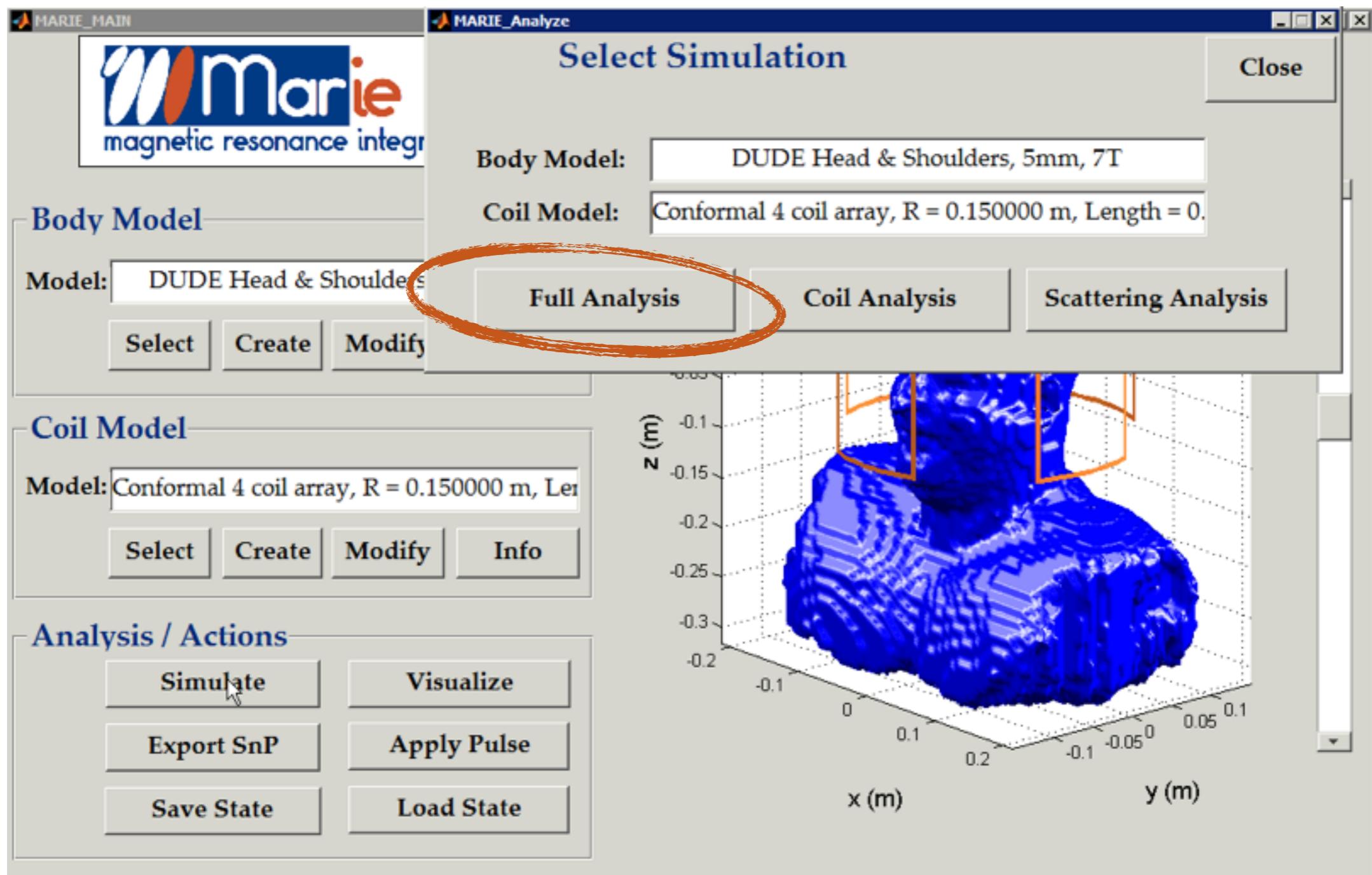


## ♦ Call the simulation engine

- ♦ opens the simulation dialogue
- ♦ coil and body model names are shown

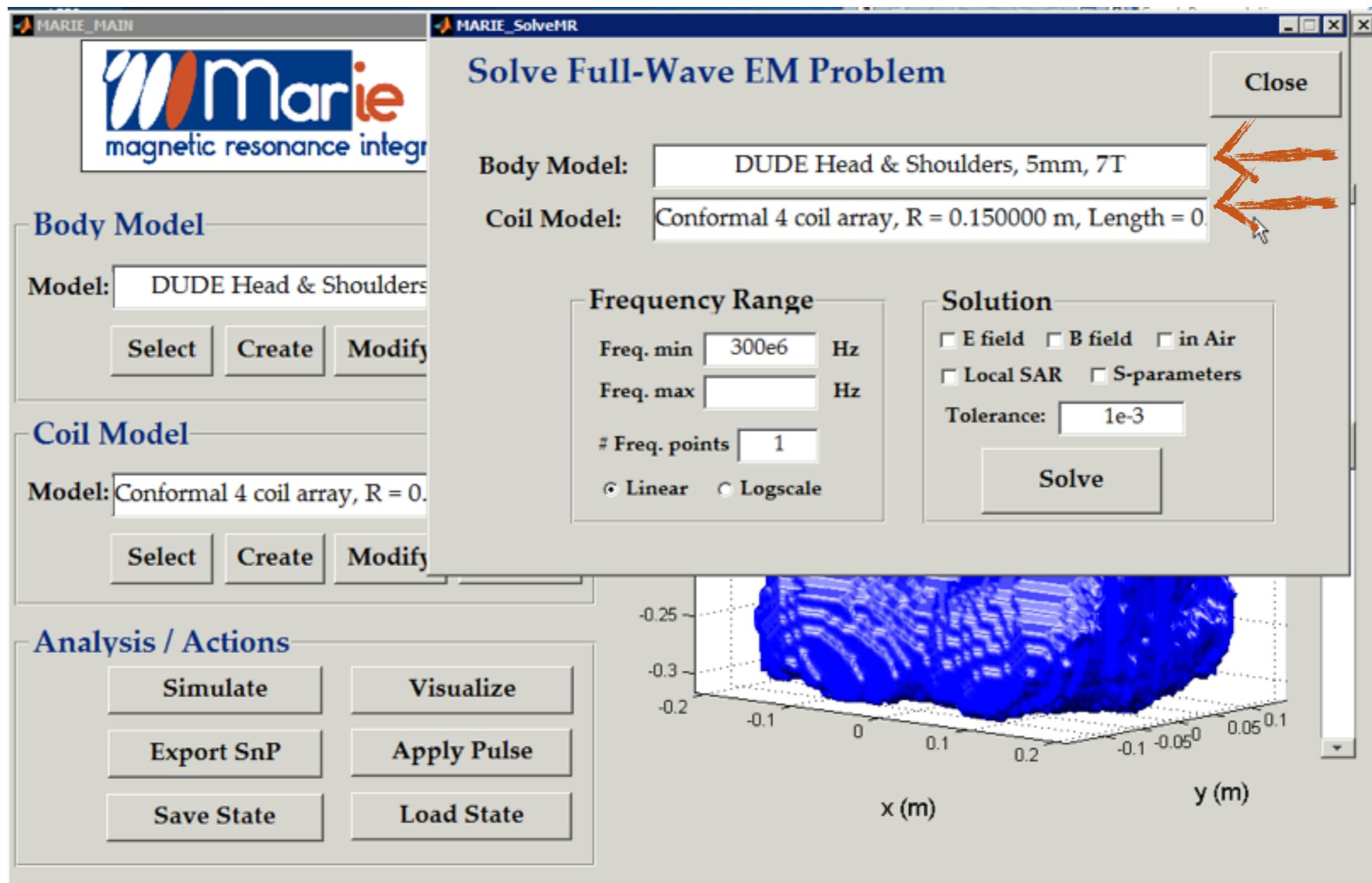


- ◆ Full-wave EM analysis
  - ◆ solves the complete problem, including body and coil models



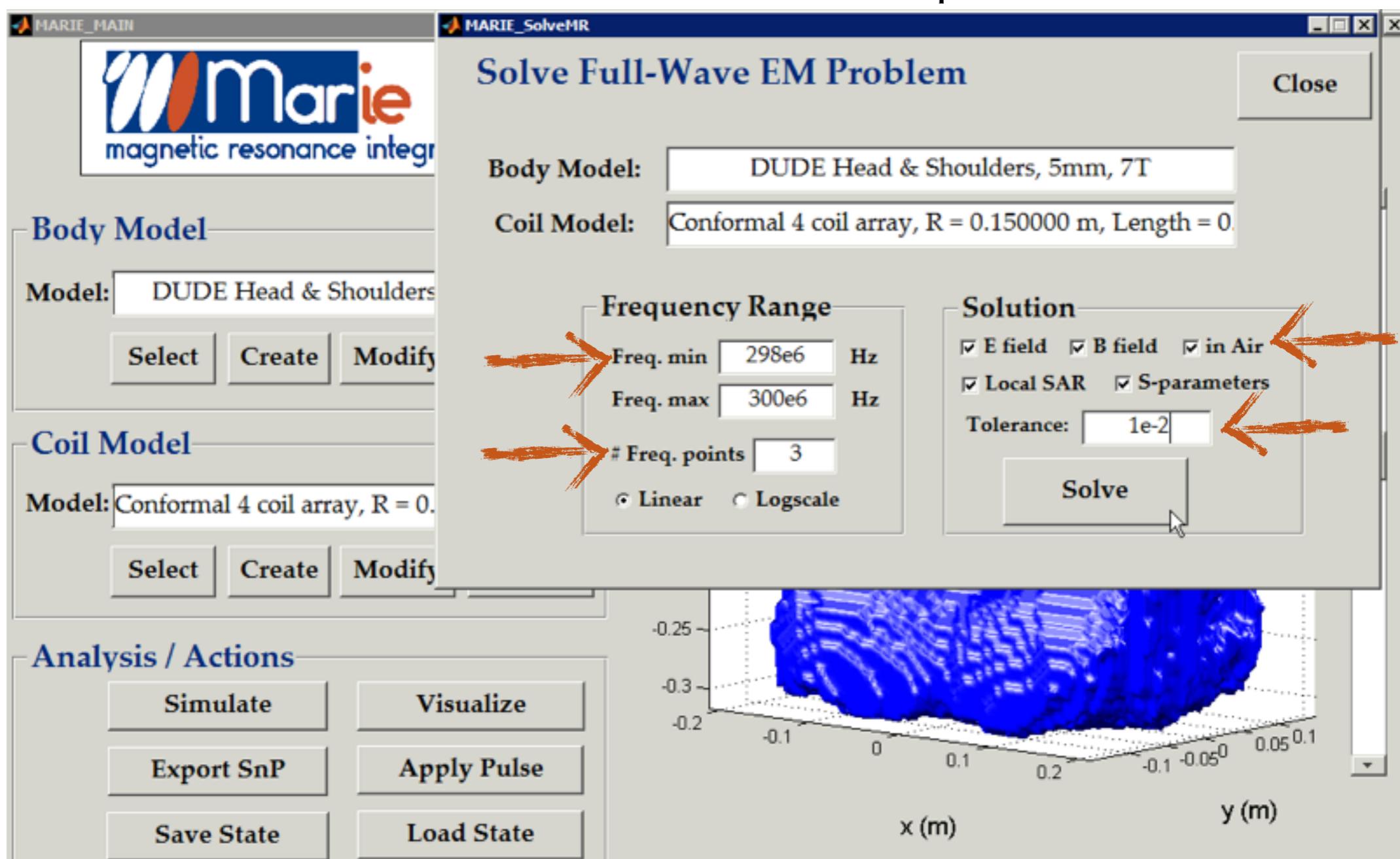
## ◆ Full-wave EM analysis

- ◆ open the MR simulation dialogue
- ◆ shows the body and coil model names



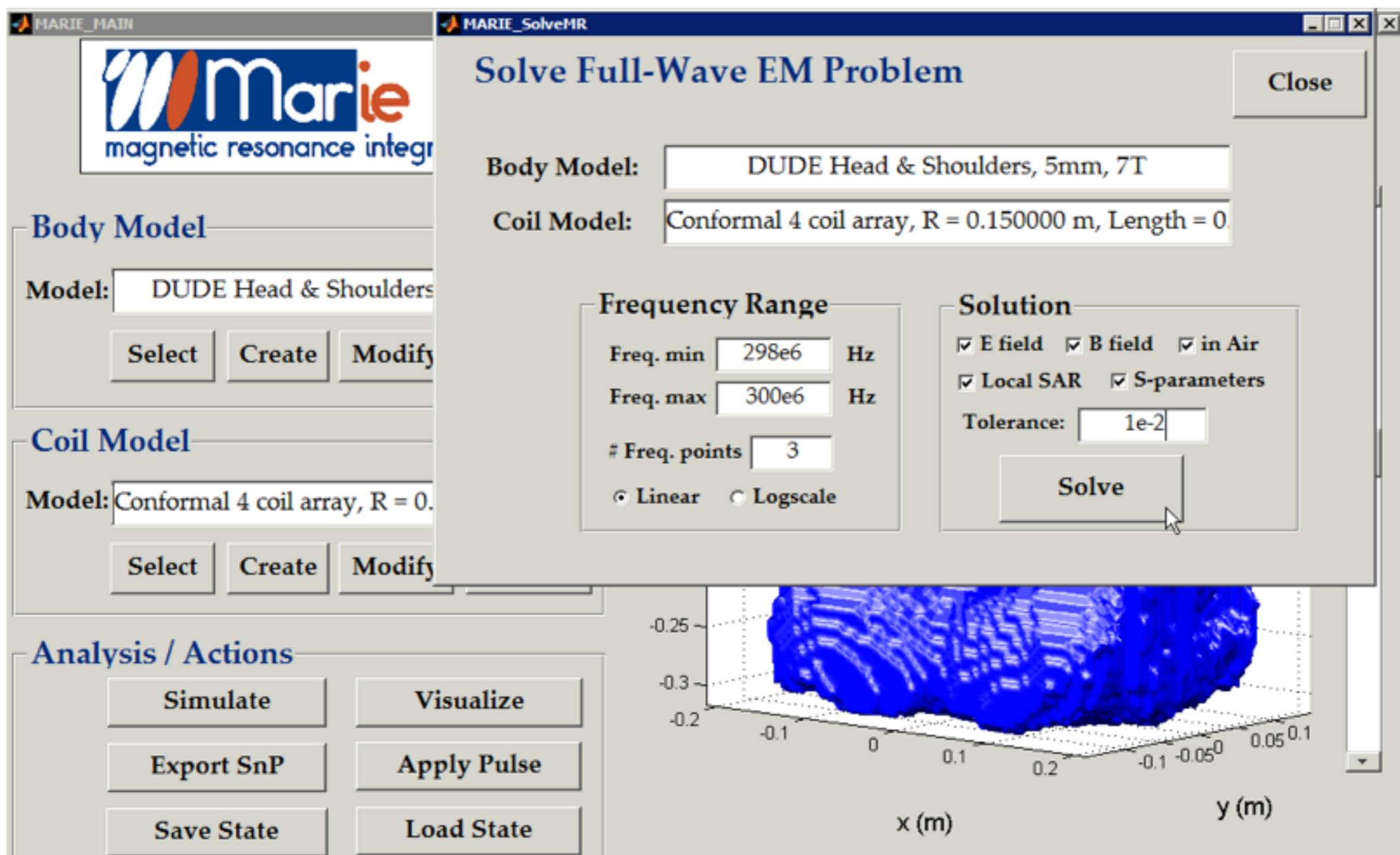
## ◆ Full-wave EM analysis

- ◆ select the frequency range and number of points
- ◆ tick the boxes of the extra data to compute
- ◆ select the error tolerance (see MARIE\_solver.pdf for details)

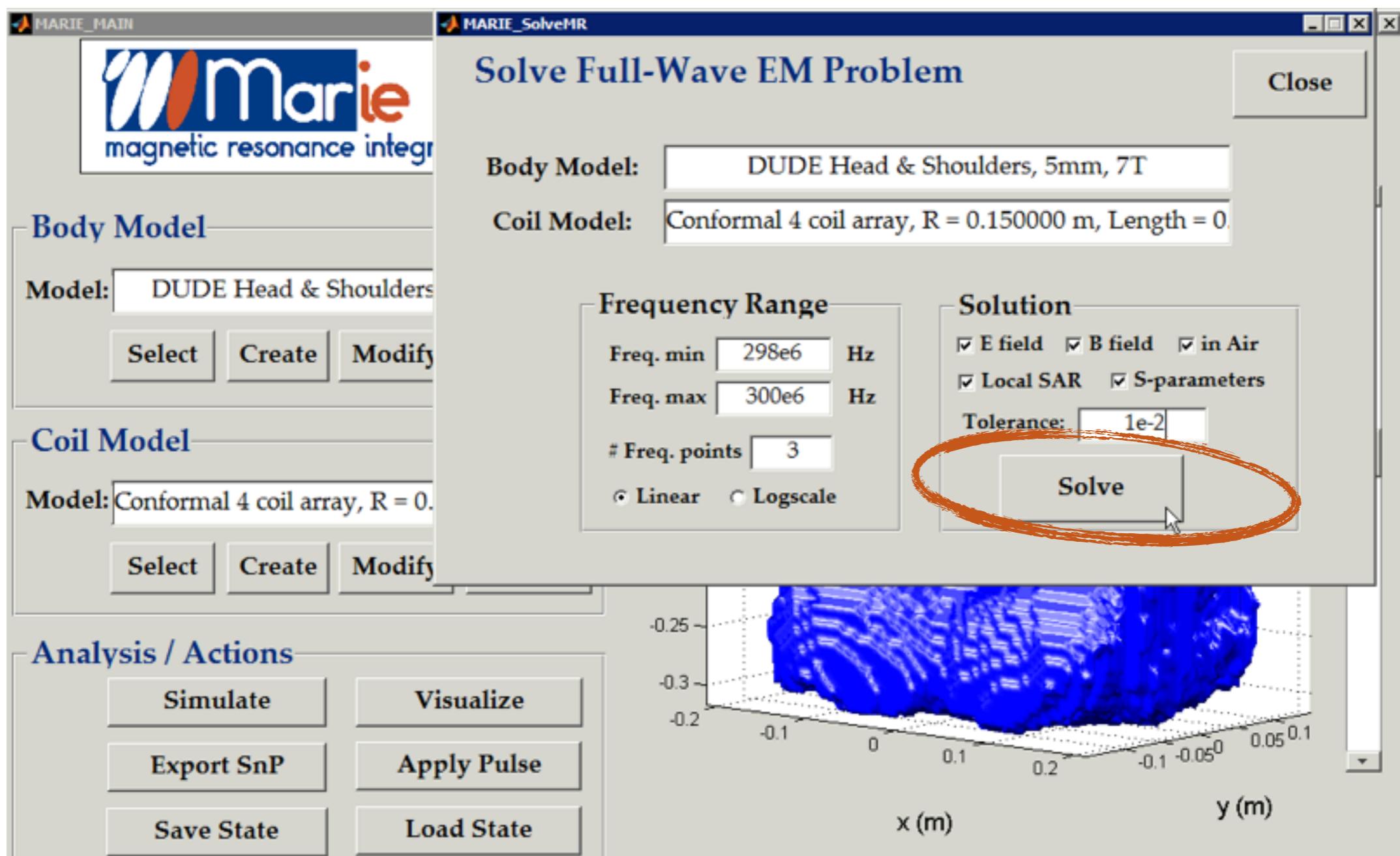


## ◆ Full-wave EM analysis

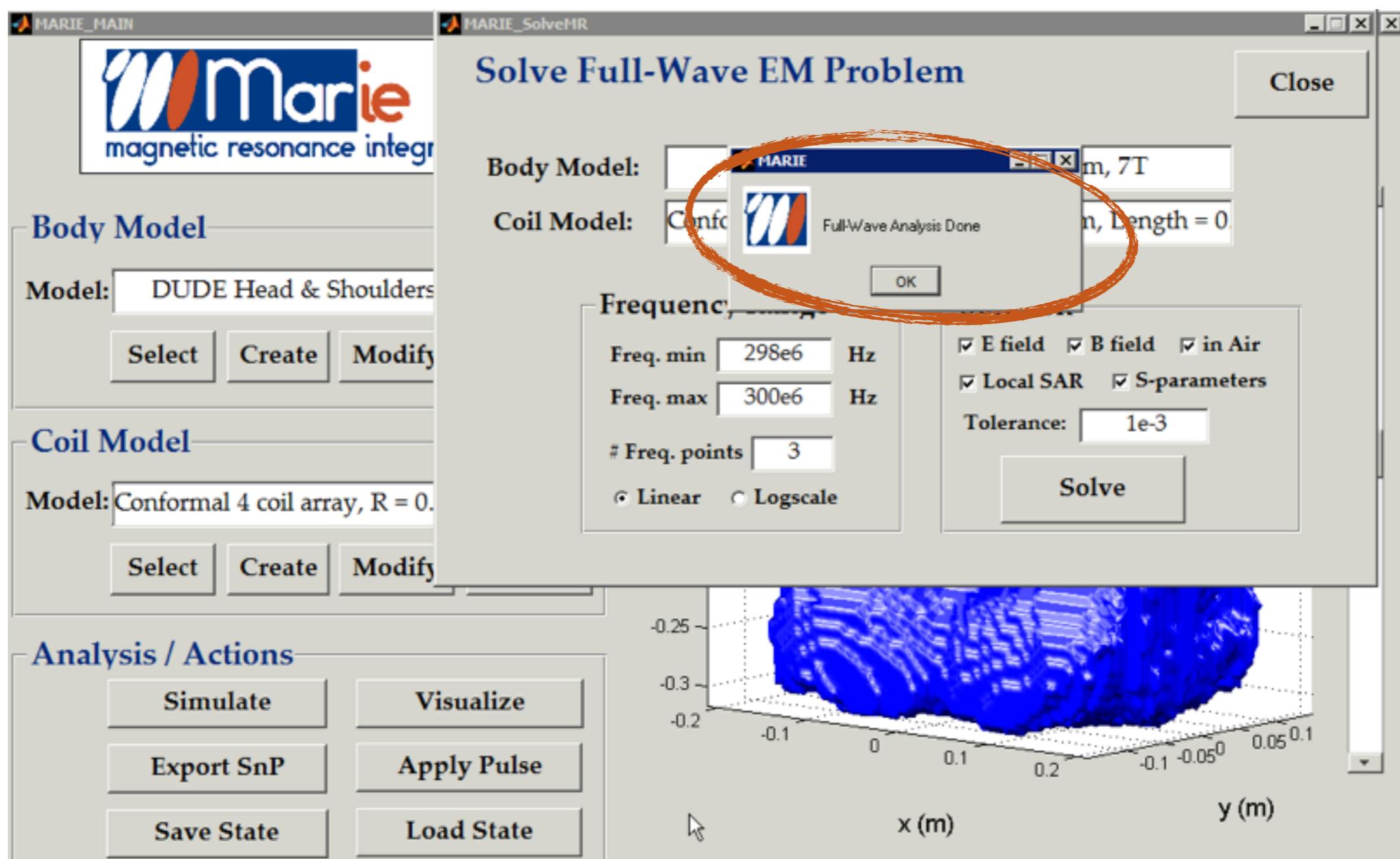
- ◆ if the “in Air” box is selected, the fields will be computed in the air
  - ◆ this may result in longer computations if the domain is large
- ◆ otherwise, the fields will only be computed inside the body



- ◆ Full-wave EM analysis
  - ◆ push “solve” when ready

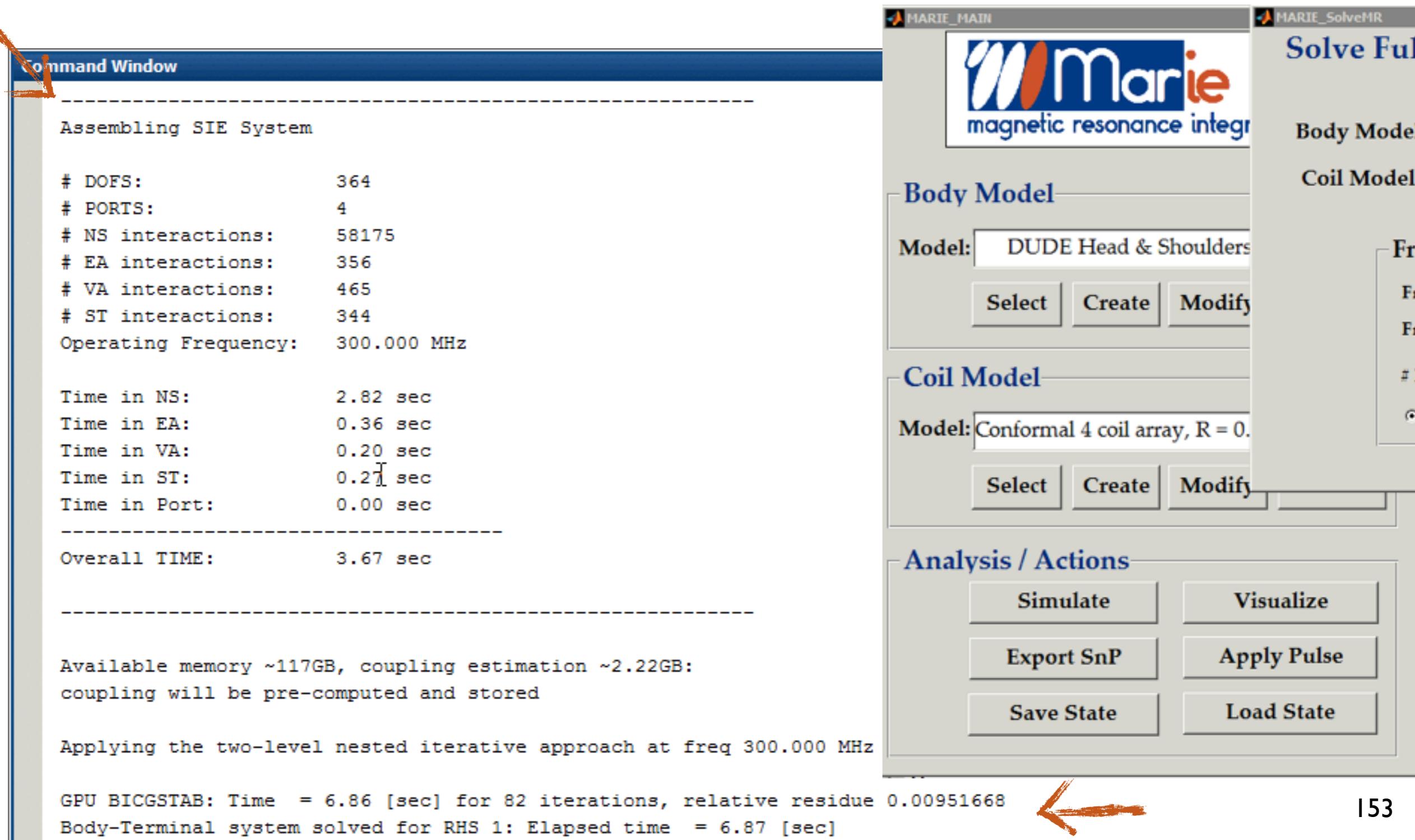


- ◆ Full-wave EM analysis
  - ◆ a window will indicate when analysis is done

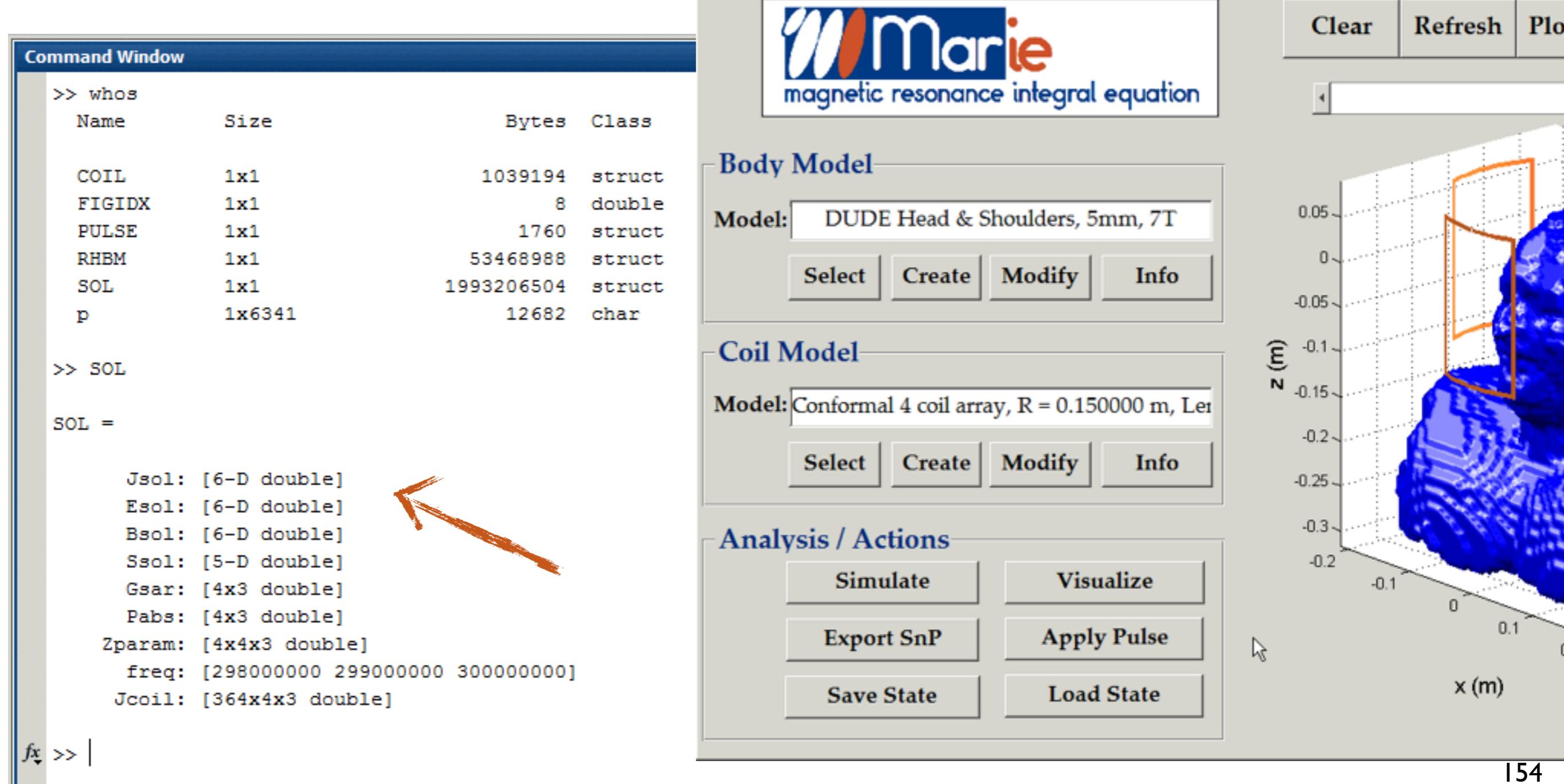


## ◆ Full-wave EM analysis

- ◆ command window will show info related to the solver



- ◆ Full-wave EM analysis
  - ◆ SOL structure is filled with the analysis results



- ◆ Full-wave EM analysis
- ◆ PULSE is still empty

**Command Window**

```
>> SOL
SOL =
    Jsol: [6-D double]
    Esol: [6-D double]
    Bsol: [6-D double]
    Ssol: [5-D double]
    Gsar: [4x3 double]
    Pabs: [4x3 double]
    Zparam: [4x4x3 double]
    freq: [298000000 299000000 300000000]
    Jcoil: [364x4x3 double]

>> PULSE
PULSE =
    Jcoil: []
    Jsol: []
    Esol: []
    Bsol: []
    Ssol: []
    Gsar: []
    Pabs: []
    Ipulse: []
    Vpulse: []
    freq: []
```

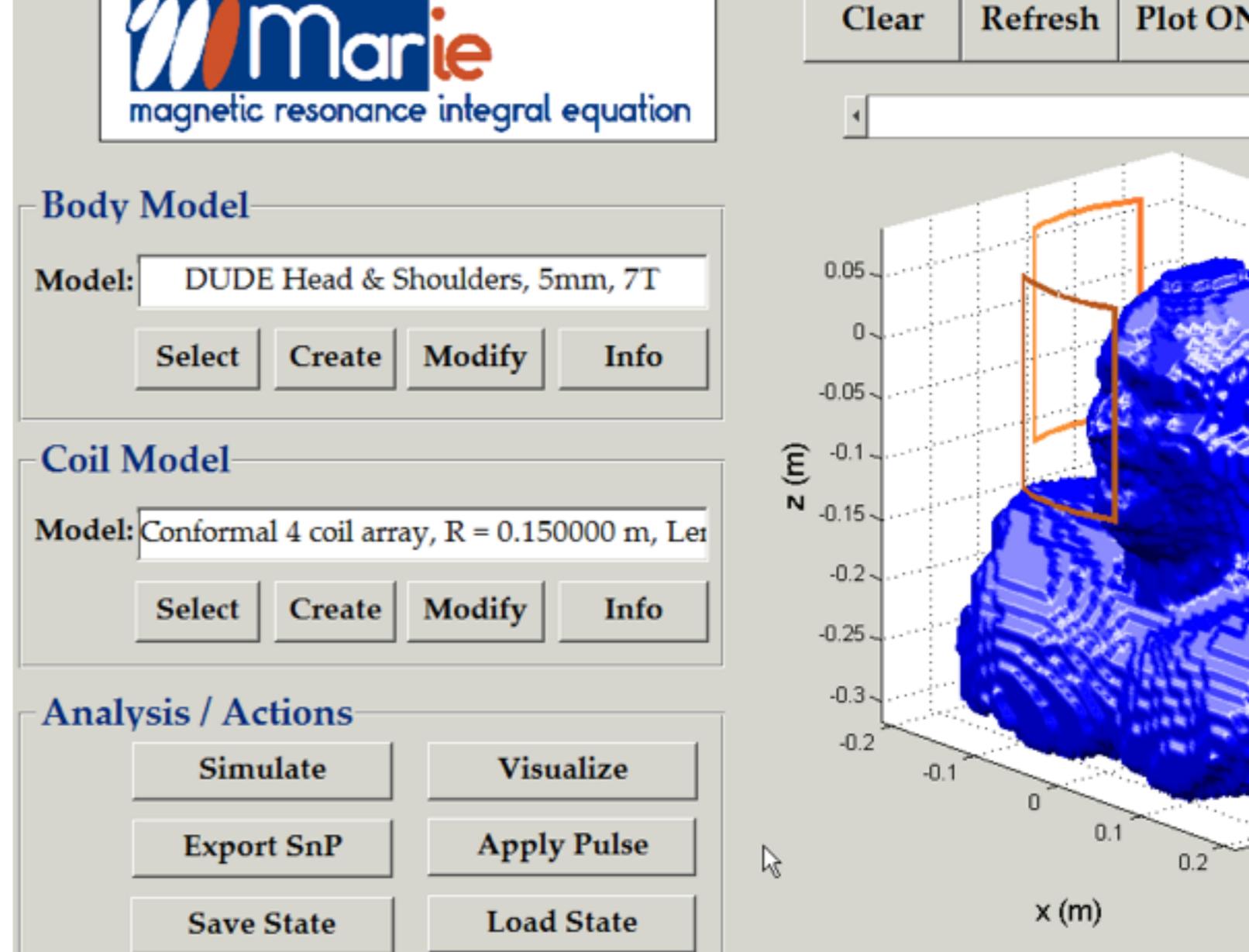
**MARIE\_MAIN**

**Marie**  
magnetic resonance integral equation

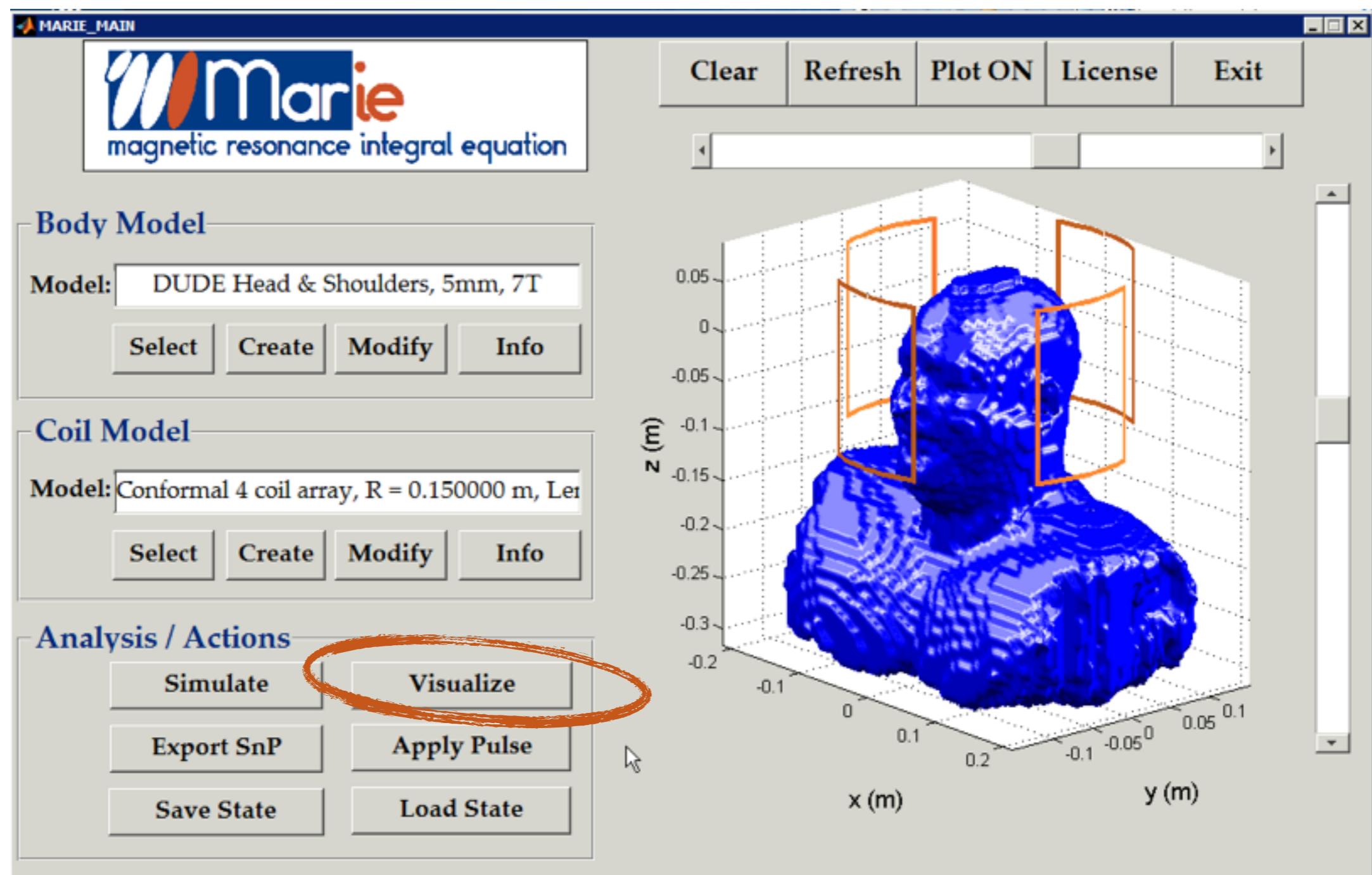
**Body Model**  
Model: DUDE Head & Shoulders, 5mm, 7T  
Select Create Modify Info

**Coil Model**  
Model: Conformal 4 coil array, R = 0.150000 m, Len  
Select Create Modify Info

**Analysis / Actions**  
Simulate Visualize  
Export SnP Apply Pulse  
Save State Load State

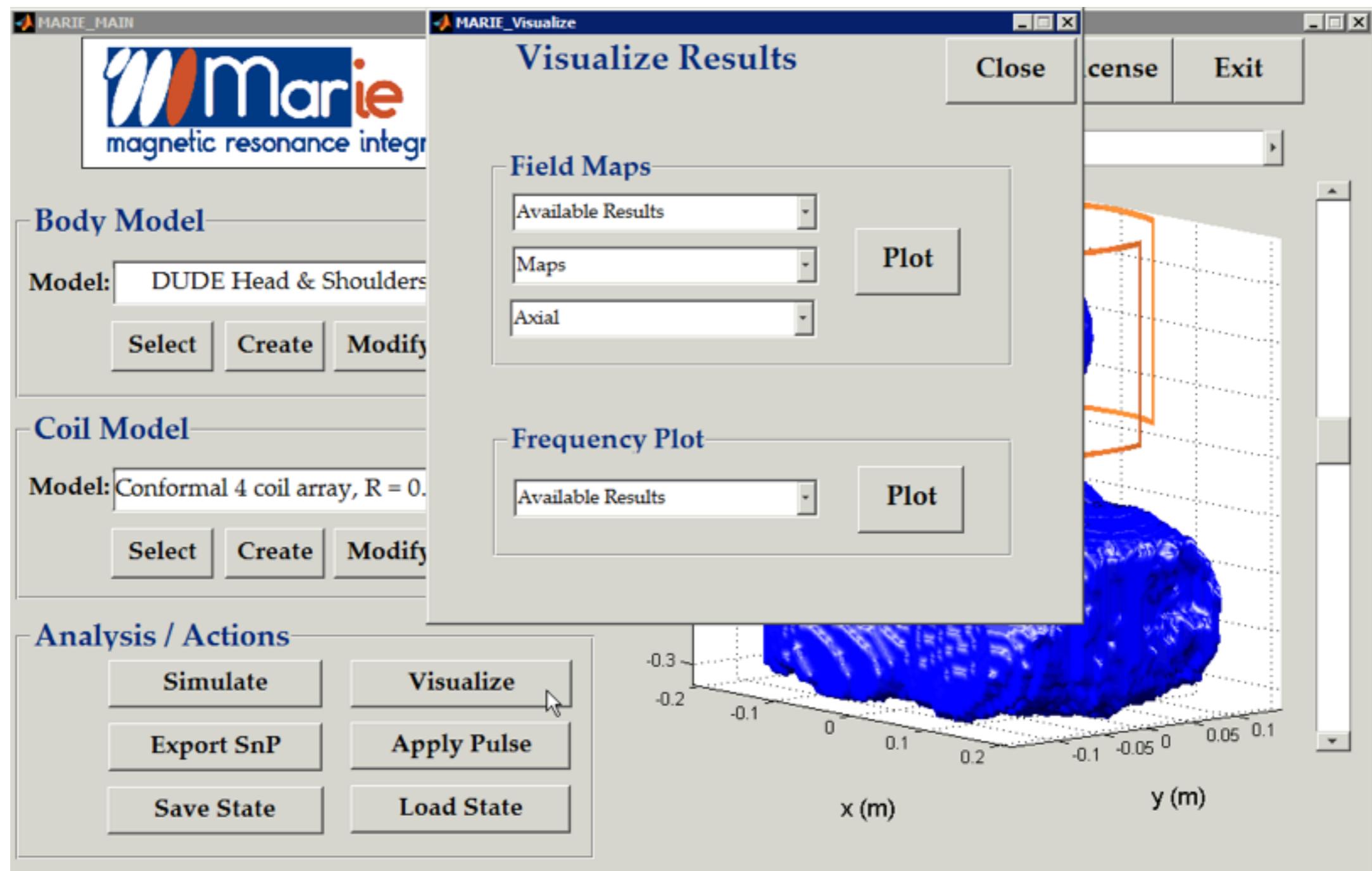


- ◆ Visualize the results
  - ◆ push the “visualize” button



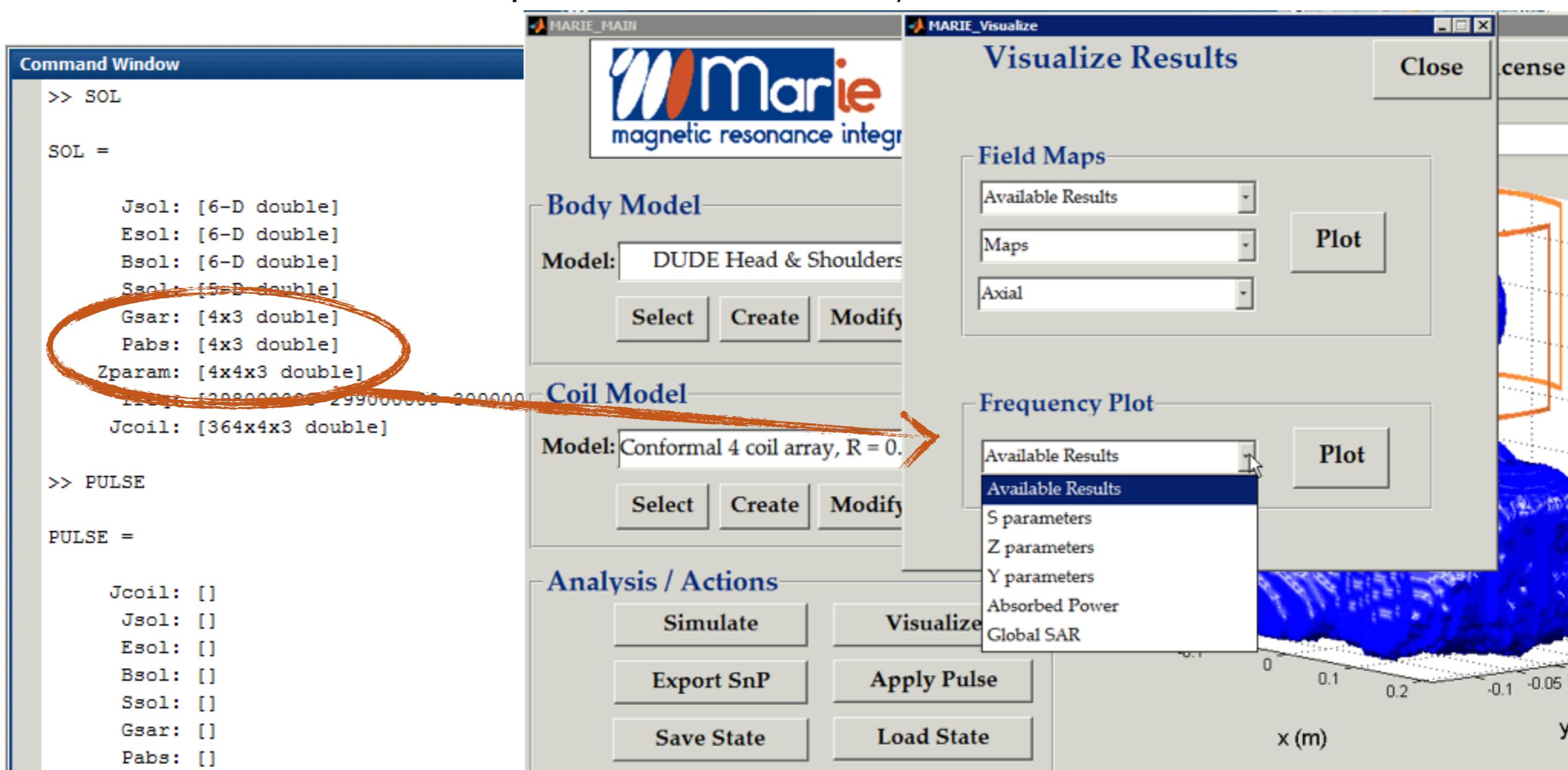
## ♦ Visualize the results

- ♦ opens the visualization dialogue box
- ♦ allows to plot field maps or frequency plots



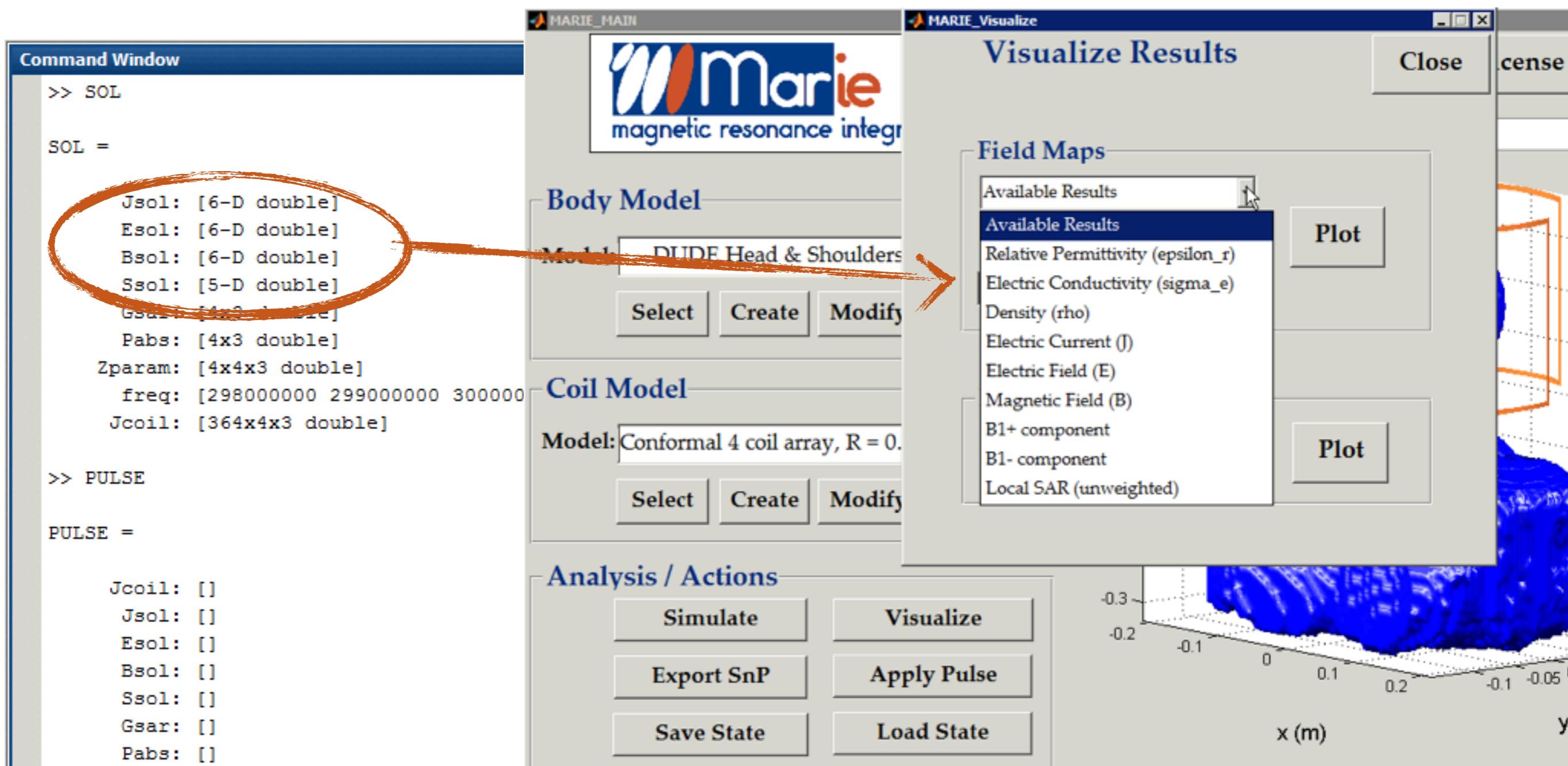
## ♦ Visualize the results

- ♦ depending on the results, there are different options available
- ♦ the port parameters include the effect of the body
  - ♦ for details of these plots, see the coil analysis below



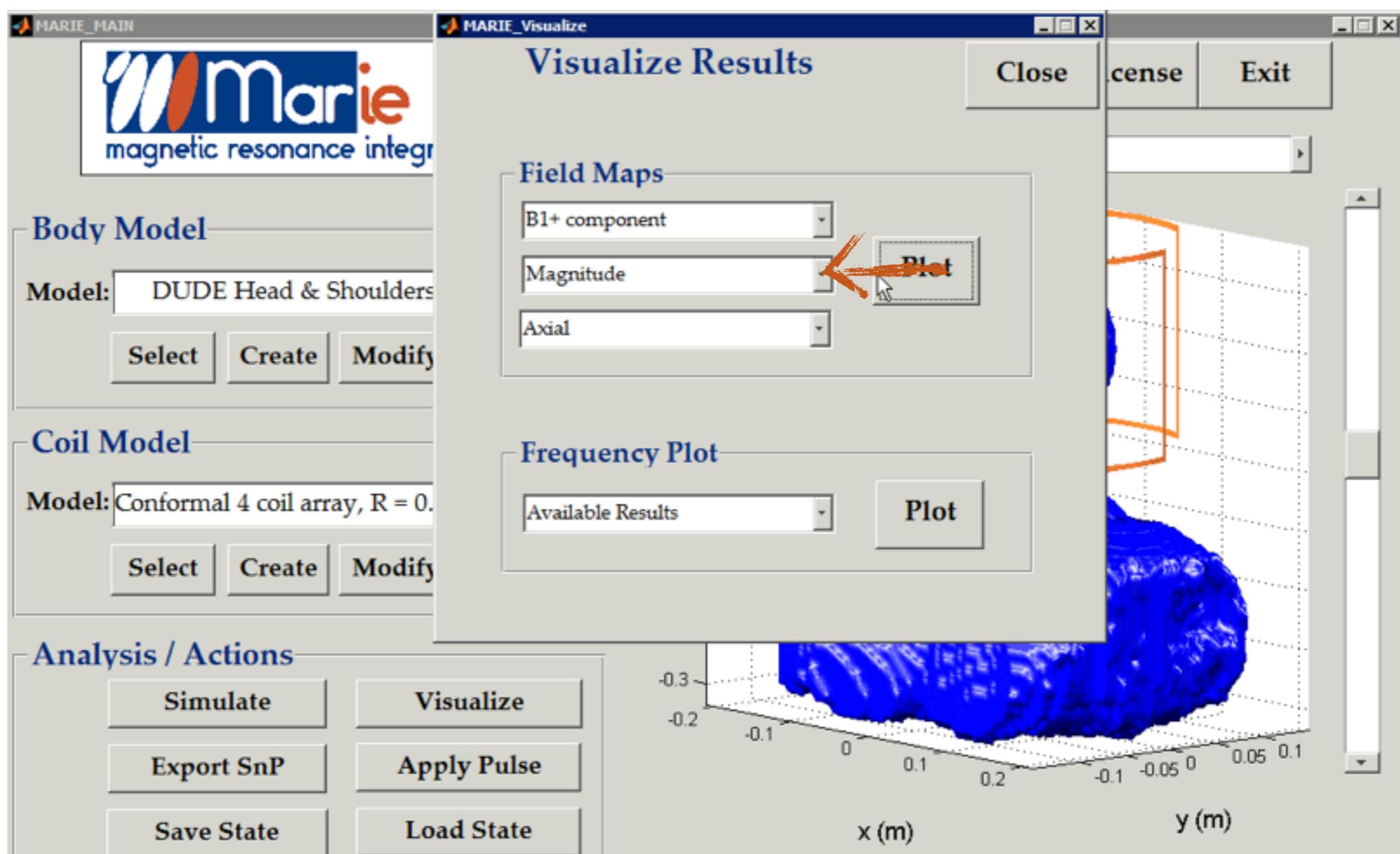
## ♦ Visualize the results

- ♦ the available results depend on the results of the simulation
- ♦ if the corresponding field box was not ticked, there is no result

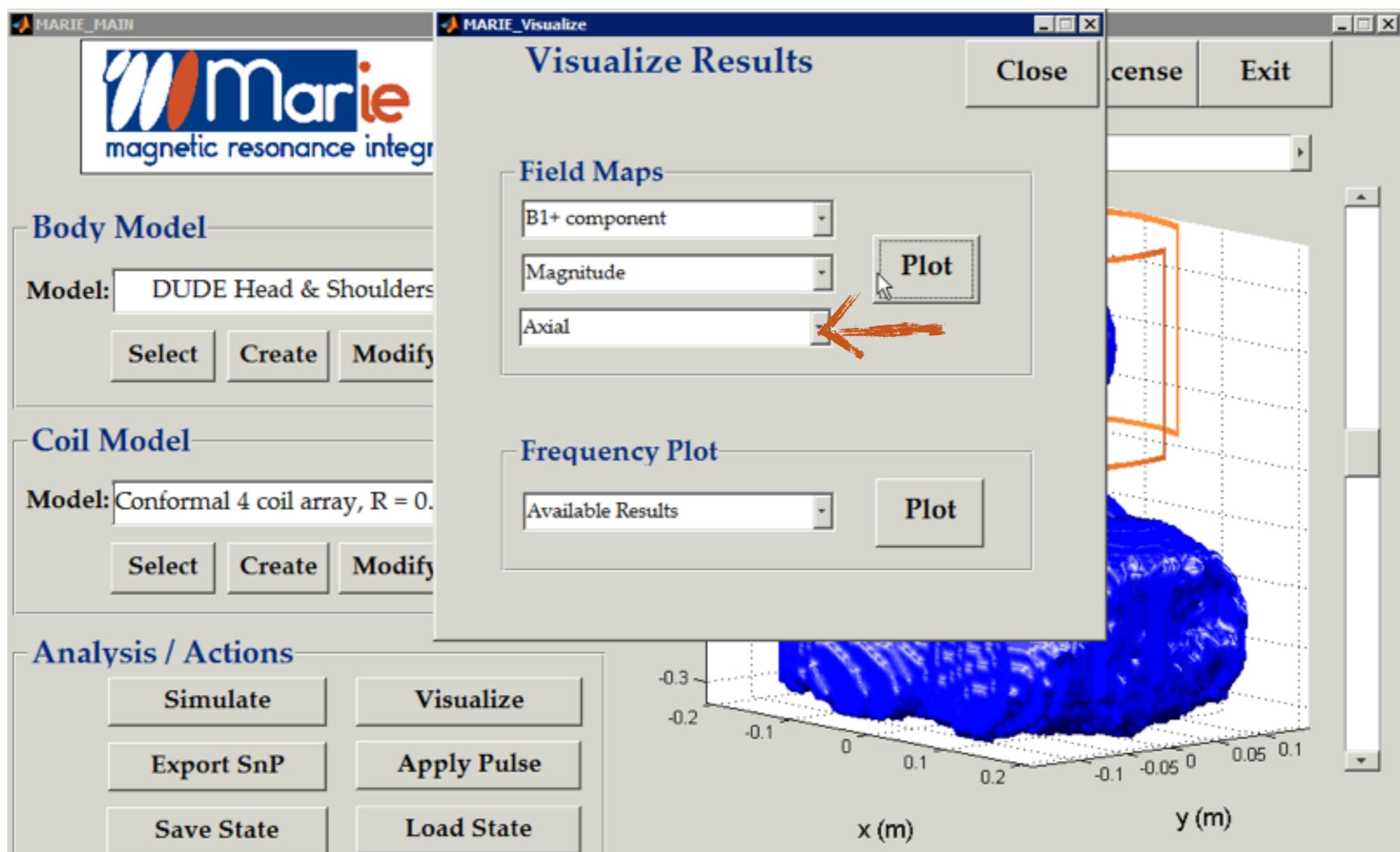


## ♦ Visualize the results

- ♦ different maps can be plotted
- ♦ depending on the results, some maps are not available
  - ♦ e.g. makes no sense to plot the components of the SAR

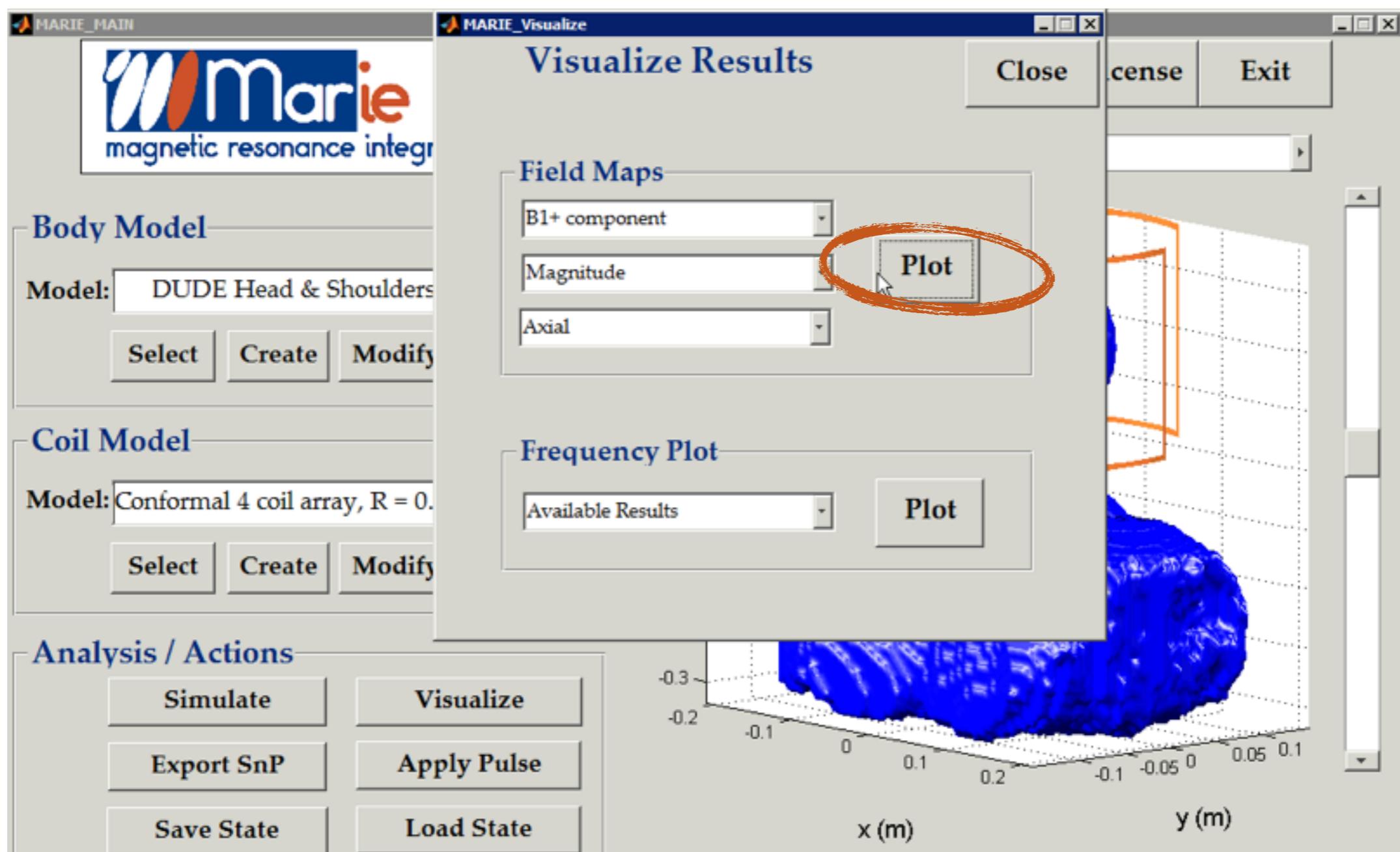


- ◆ Visualize the results
  - ◆ the slice map orientation can be selected



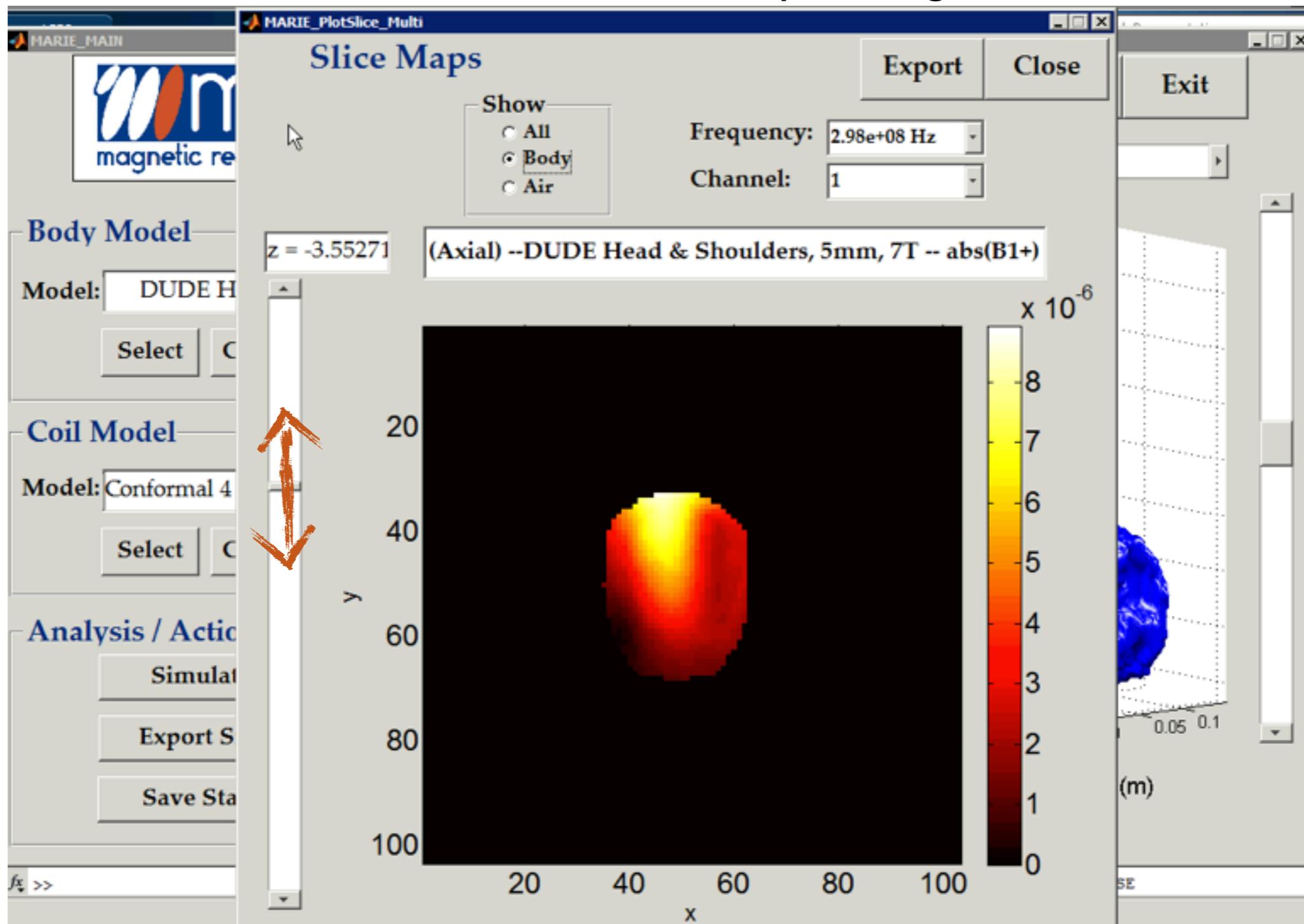
## ♦ Visualize the results

- ♦ select the desired map
- ♦ push “plot” button to open the map figure



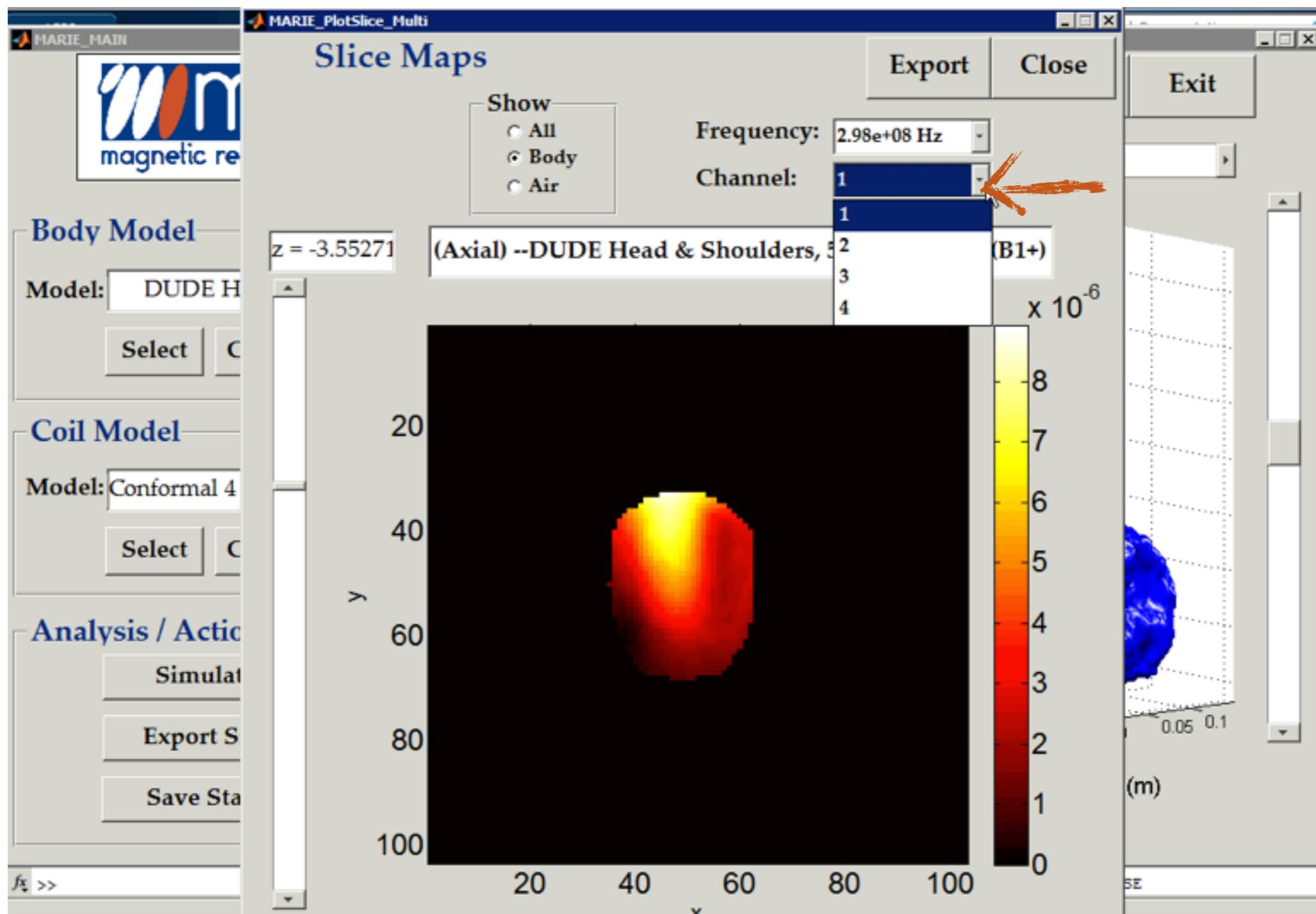
## ♦ Visualize the results

- ♦ the slice map utility opens
- ♦ the slide at the side controls the corresponding coordinate of the cut



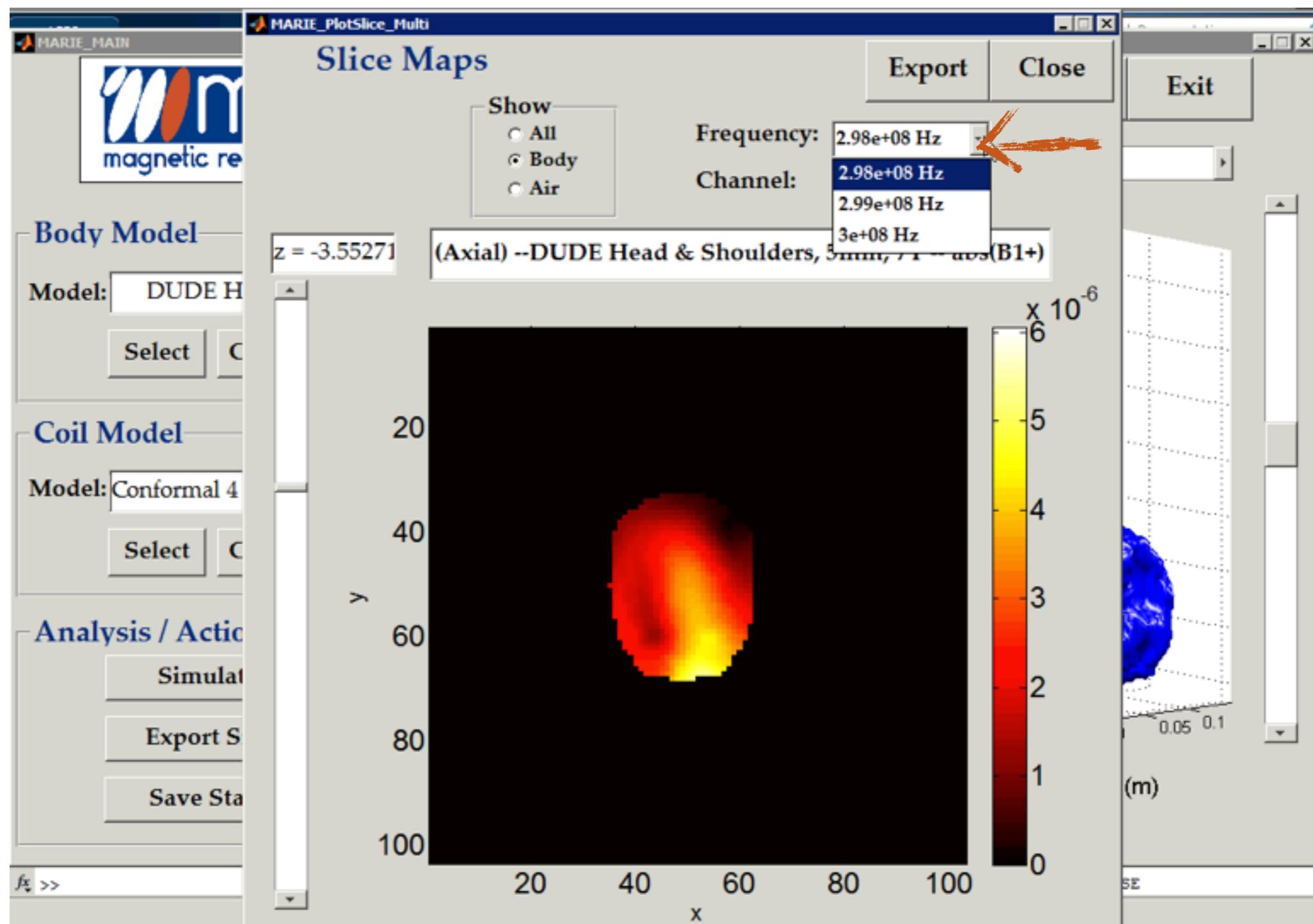
## ♦ Visualize the results

- ♦ we can select the map due to each channel
  - ♦ before tuning, each port or gap is treated as a channel

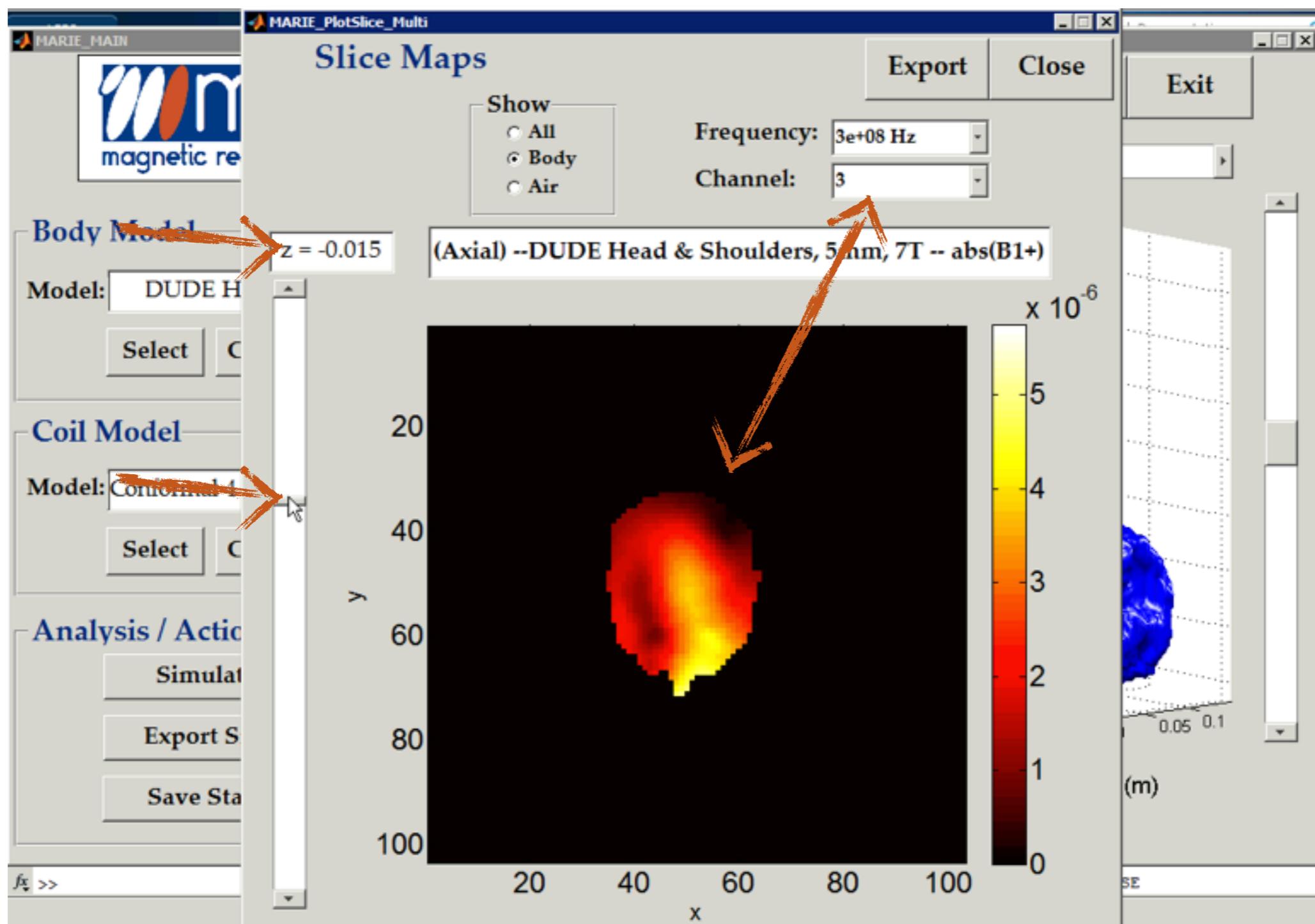


## ♦ Visualize the results

- ♦ we can select the map at the different frequencies of the simulation

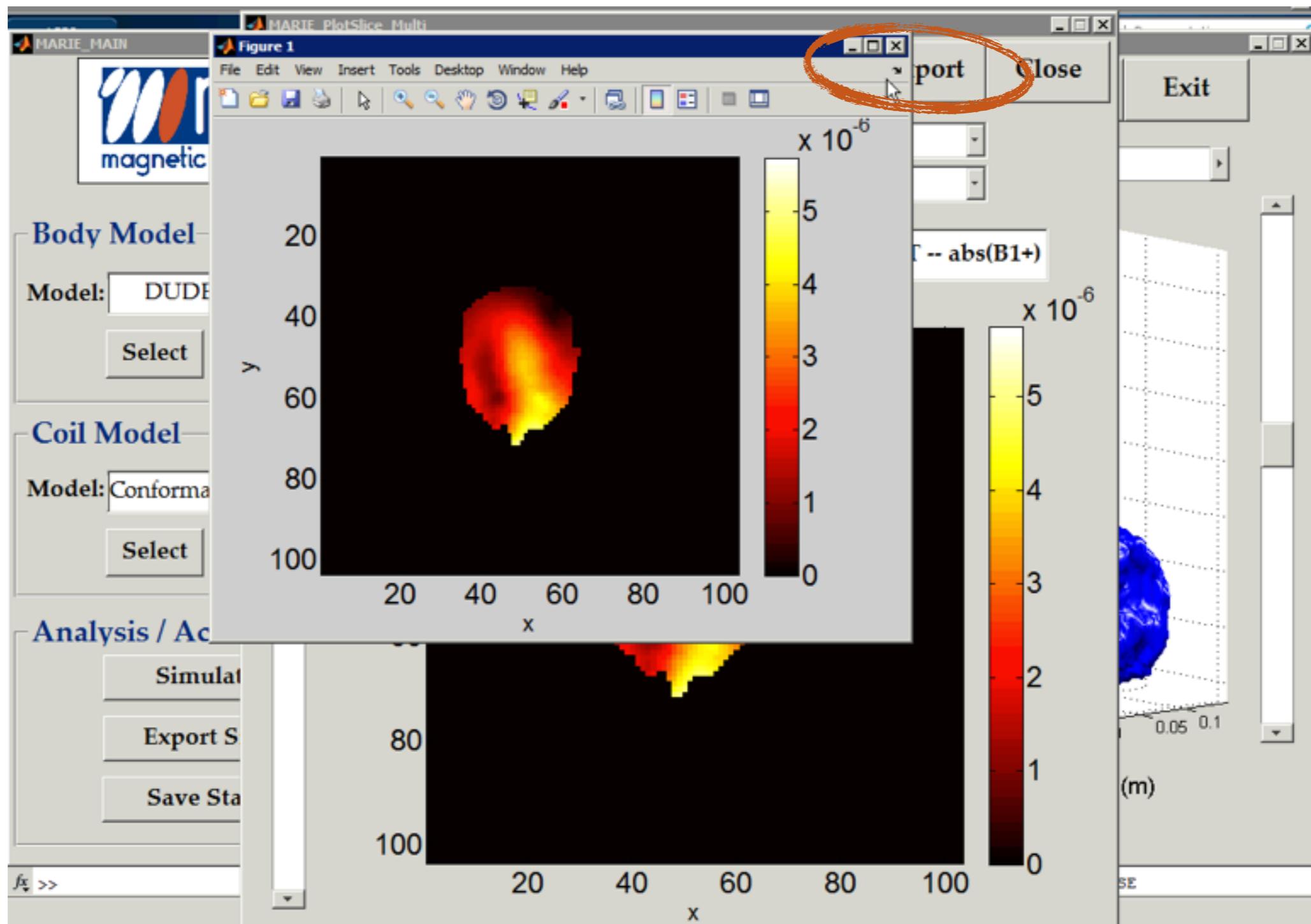


- ◆ Visualize the results
  - ◆ change to the new channel and frequency, and cut position

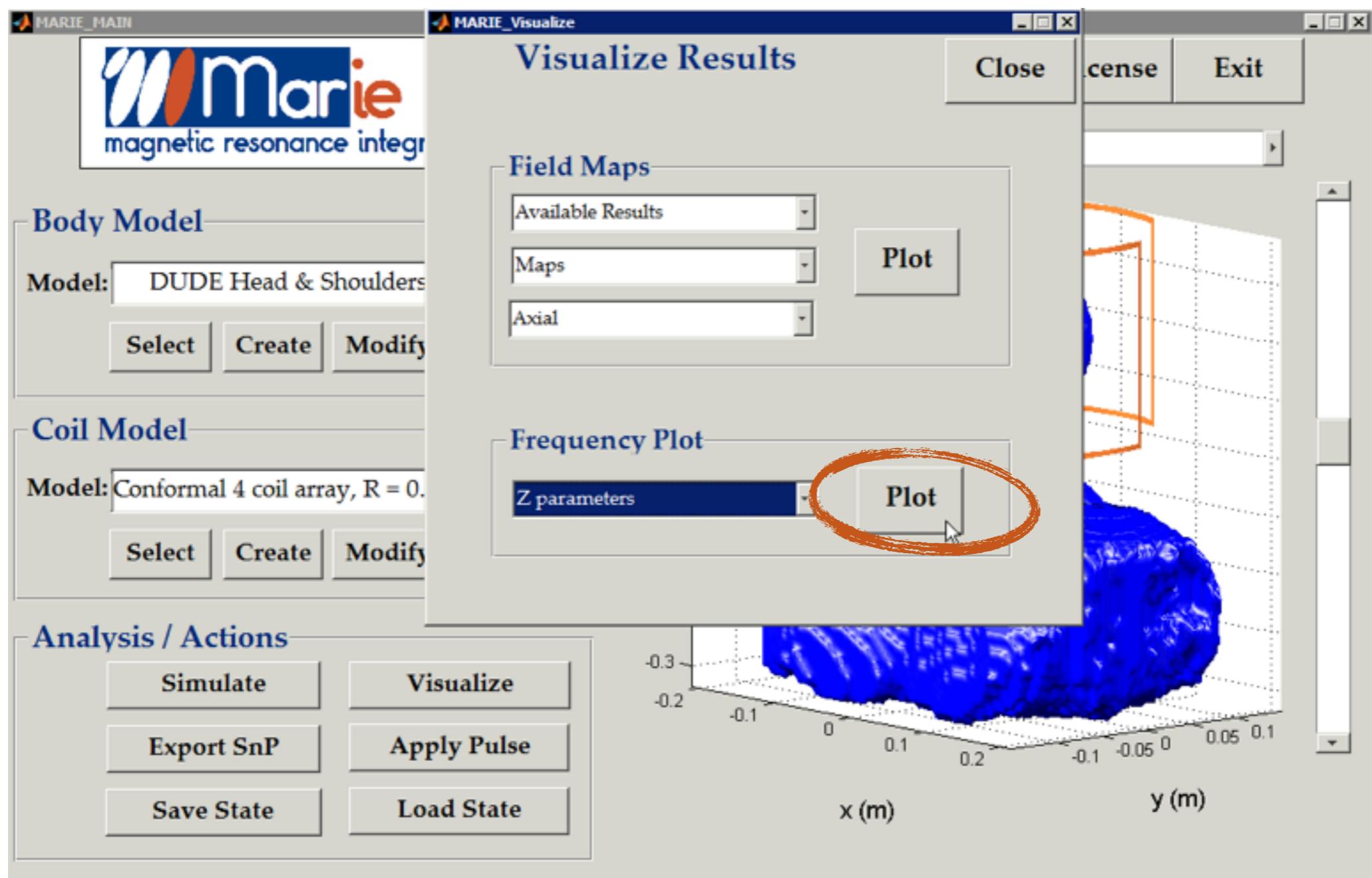


## ♦ Visualize the results

- ♦ the “export” button allows to generate a MATLAB figure of the map
  - ♦ can be edited, saved, etc...

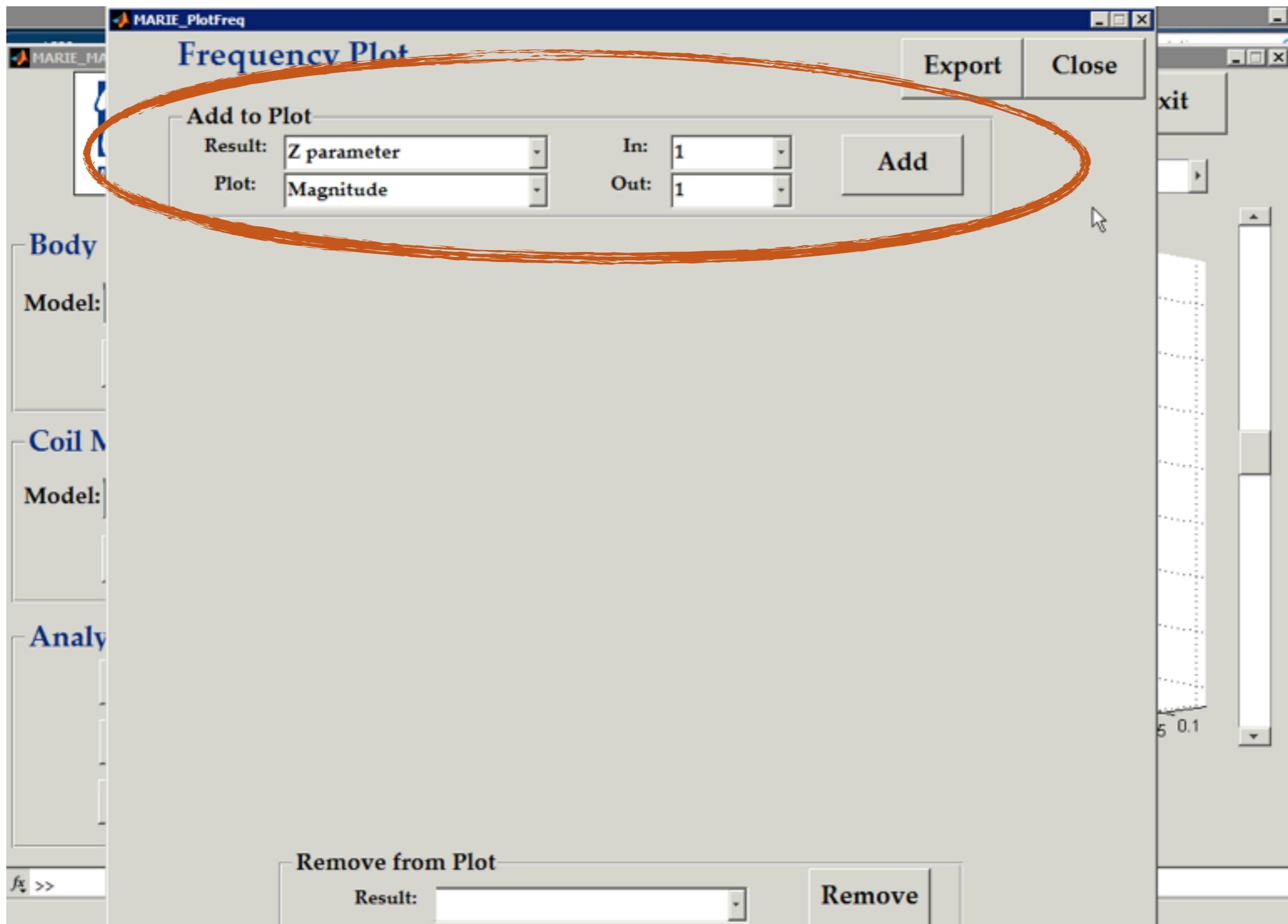


- ◆ Visualize the results
  - ◆ select result and push “plot” to call the frequency plot window

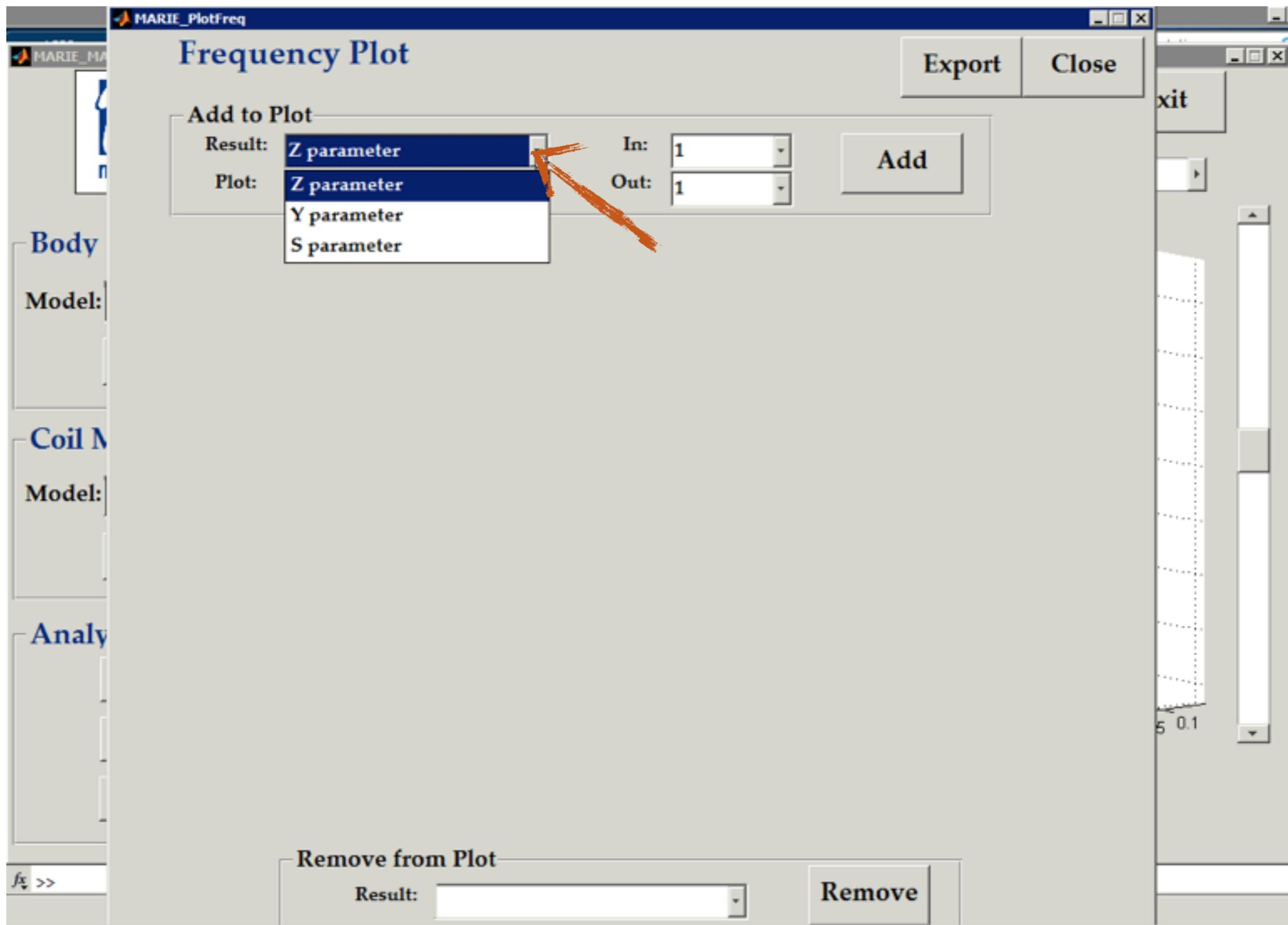


- ♦ **Visualize the results**

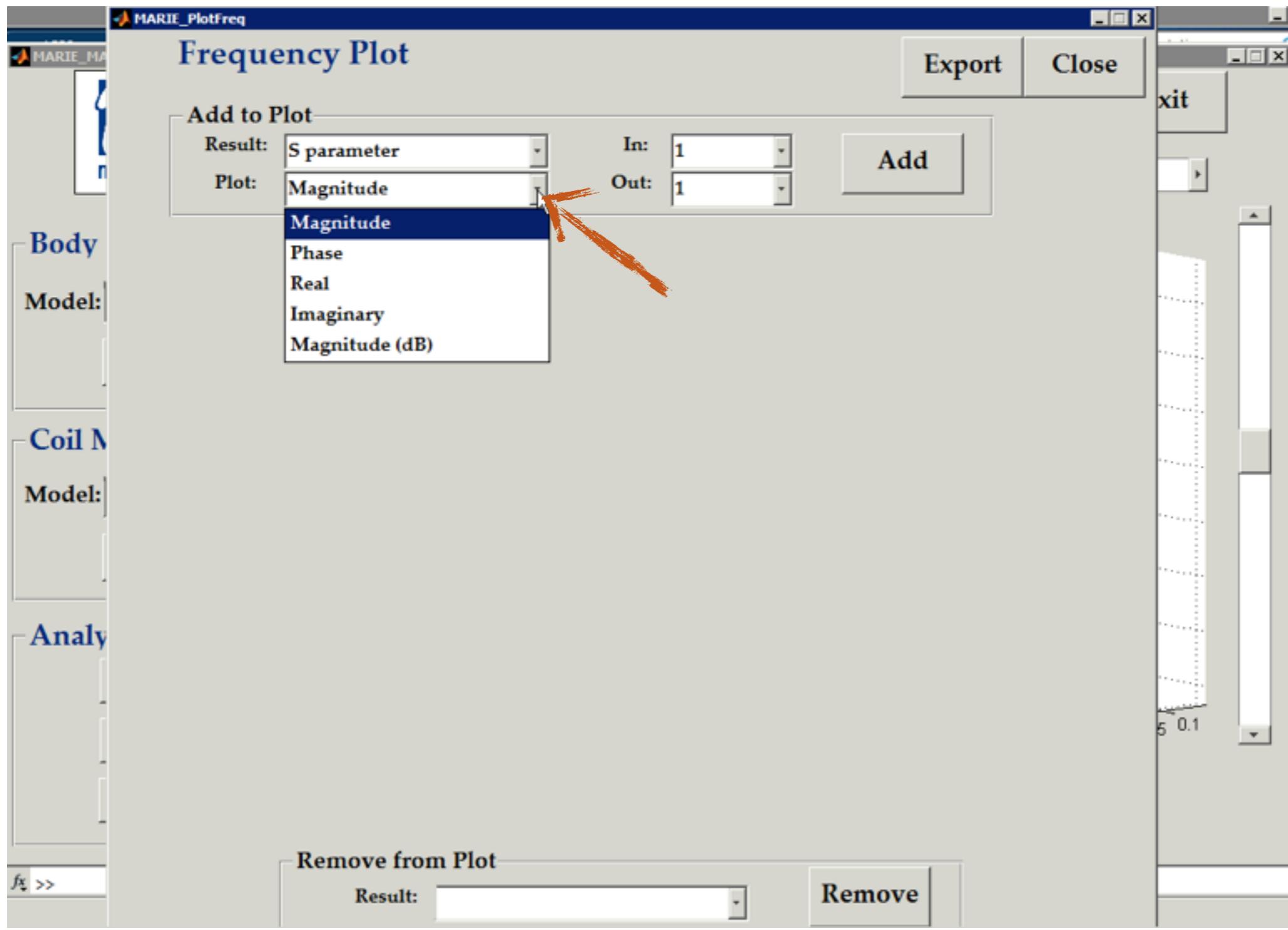
- ♦ the frequency plot application allows to select what to plot



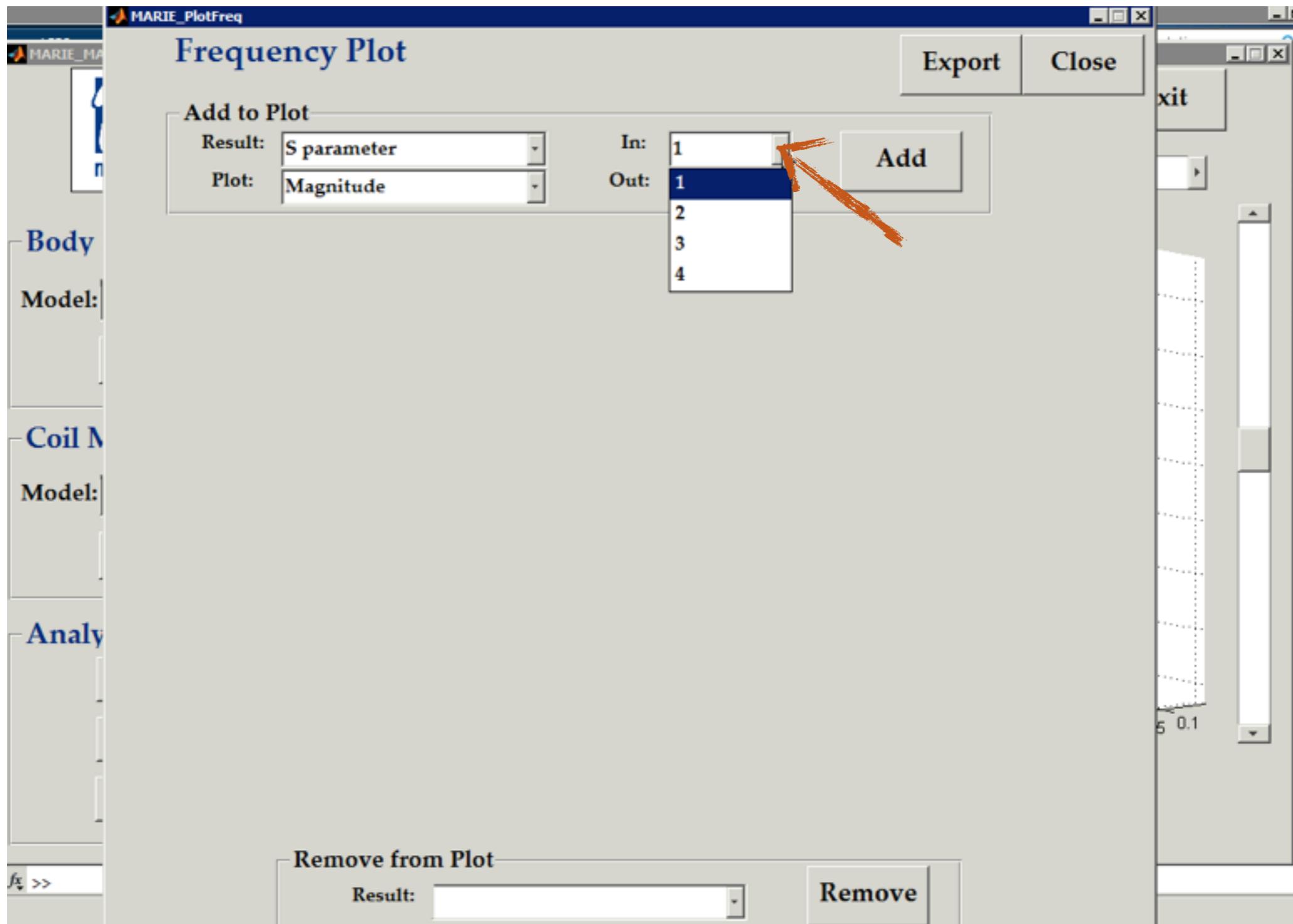
- ◆ Visualize the results
  - ◆ select figure of merit



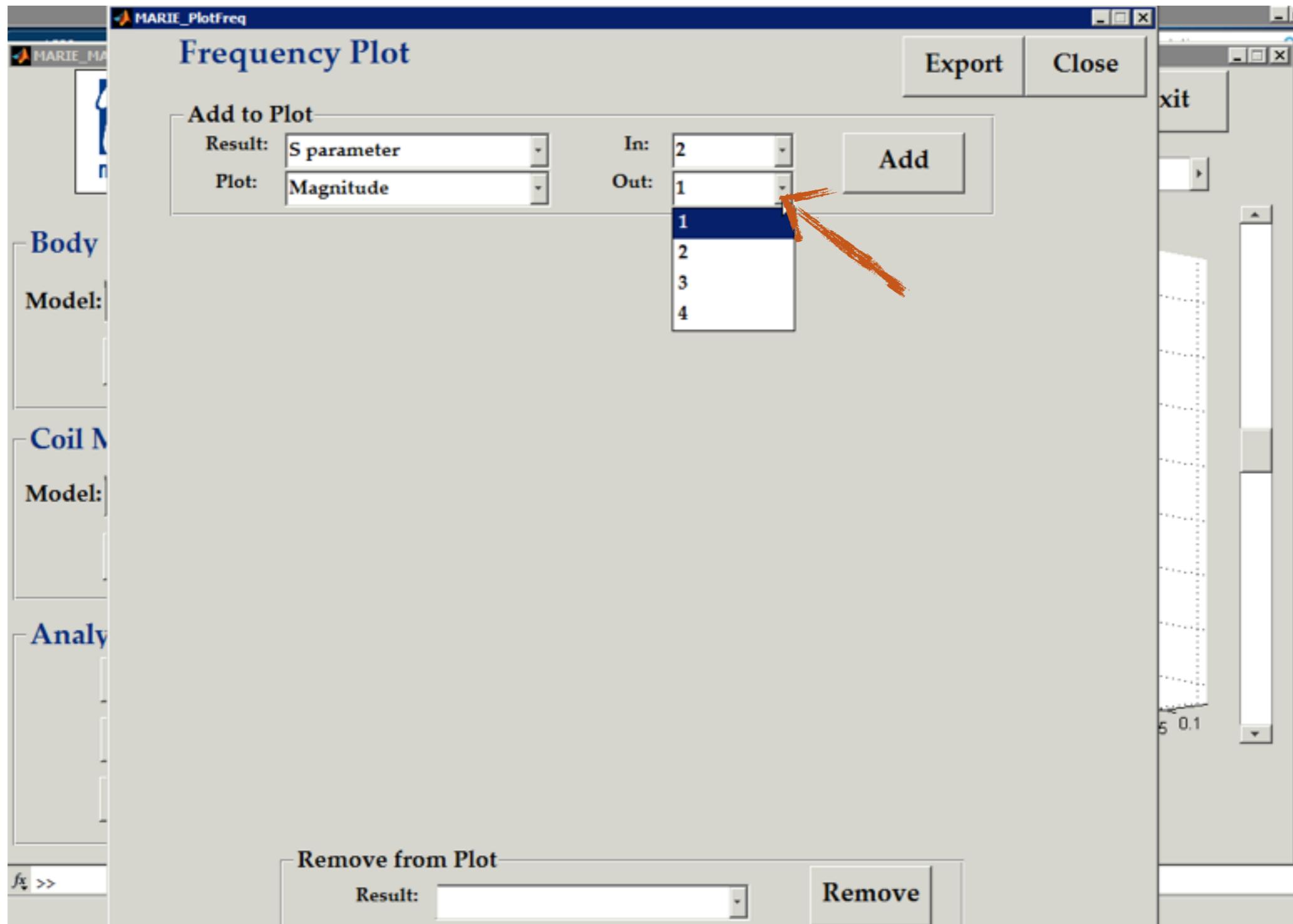
- ◆ Visualize the results
  - ◆ corresponding plot



- ◆ Visualize the results
  - ◆ input port (or channel)

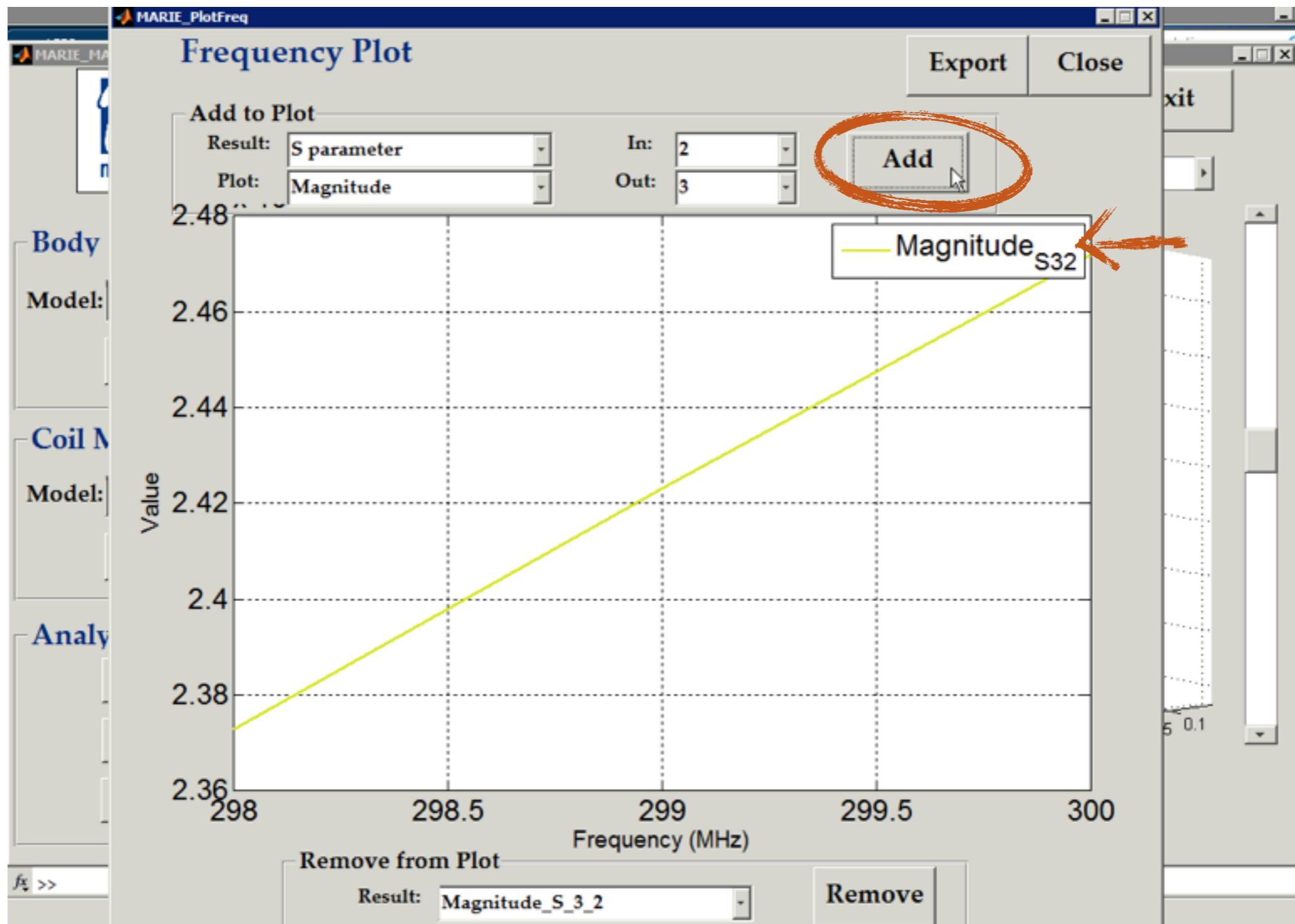


- ◆ Visualize the results
  - ◆ output port (or channel)



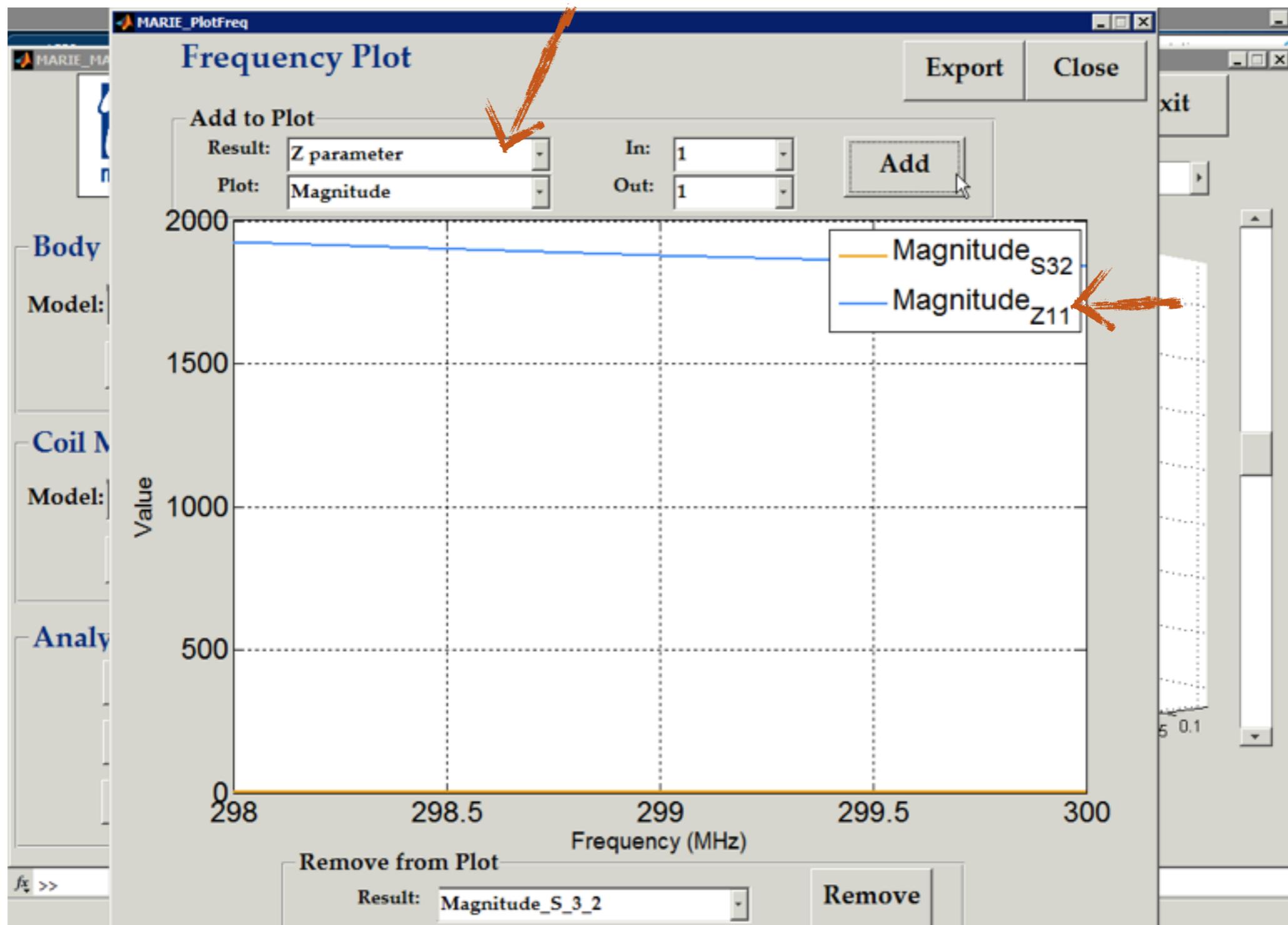
- ♦ Visualize the results

- ♦ push “add” button to add figure to the plot



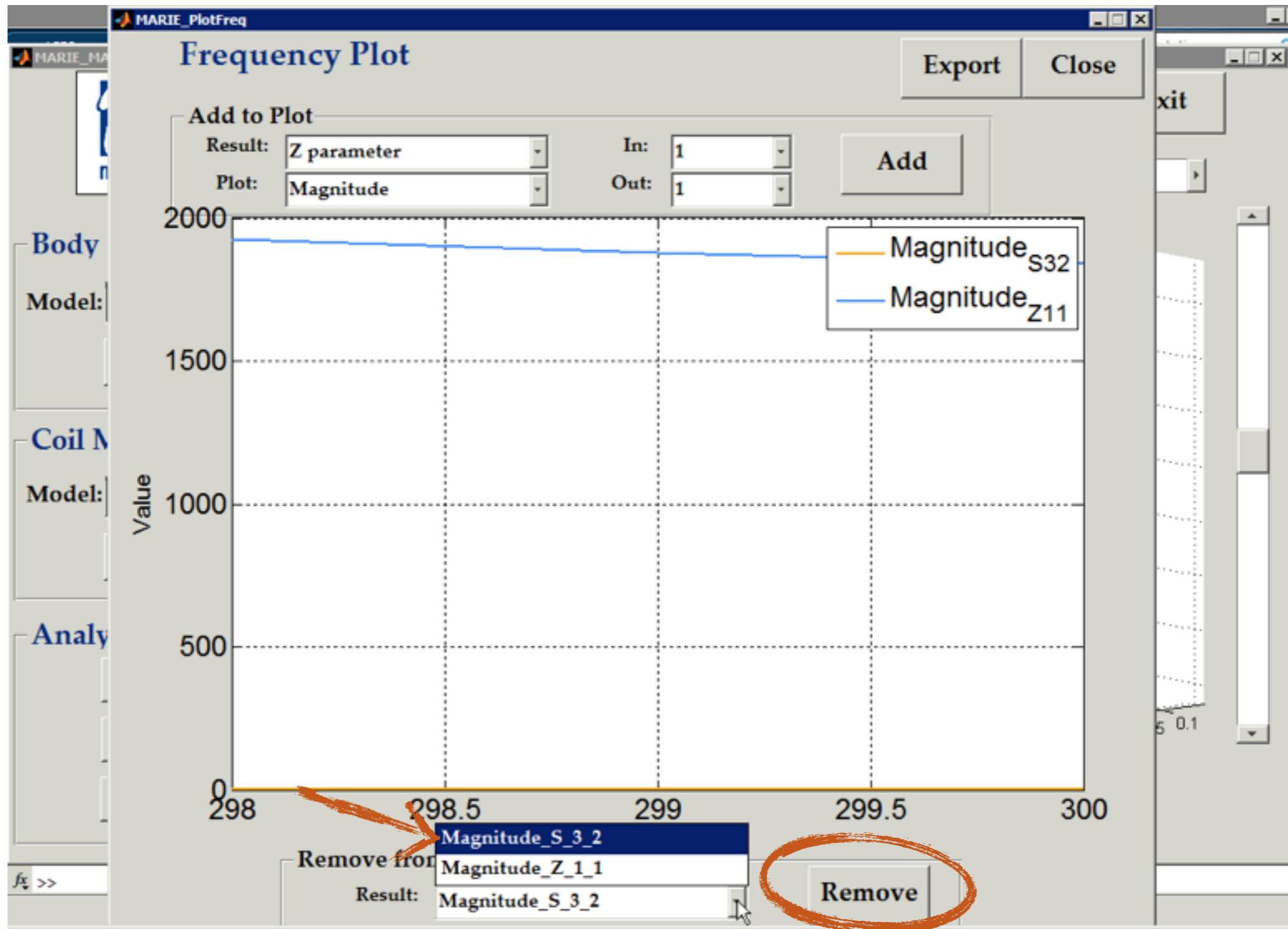
## ♦ Visualize the results

- ♦ you can add as many curves as desired, even different parameters

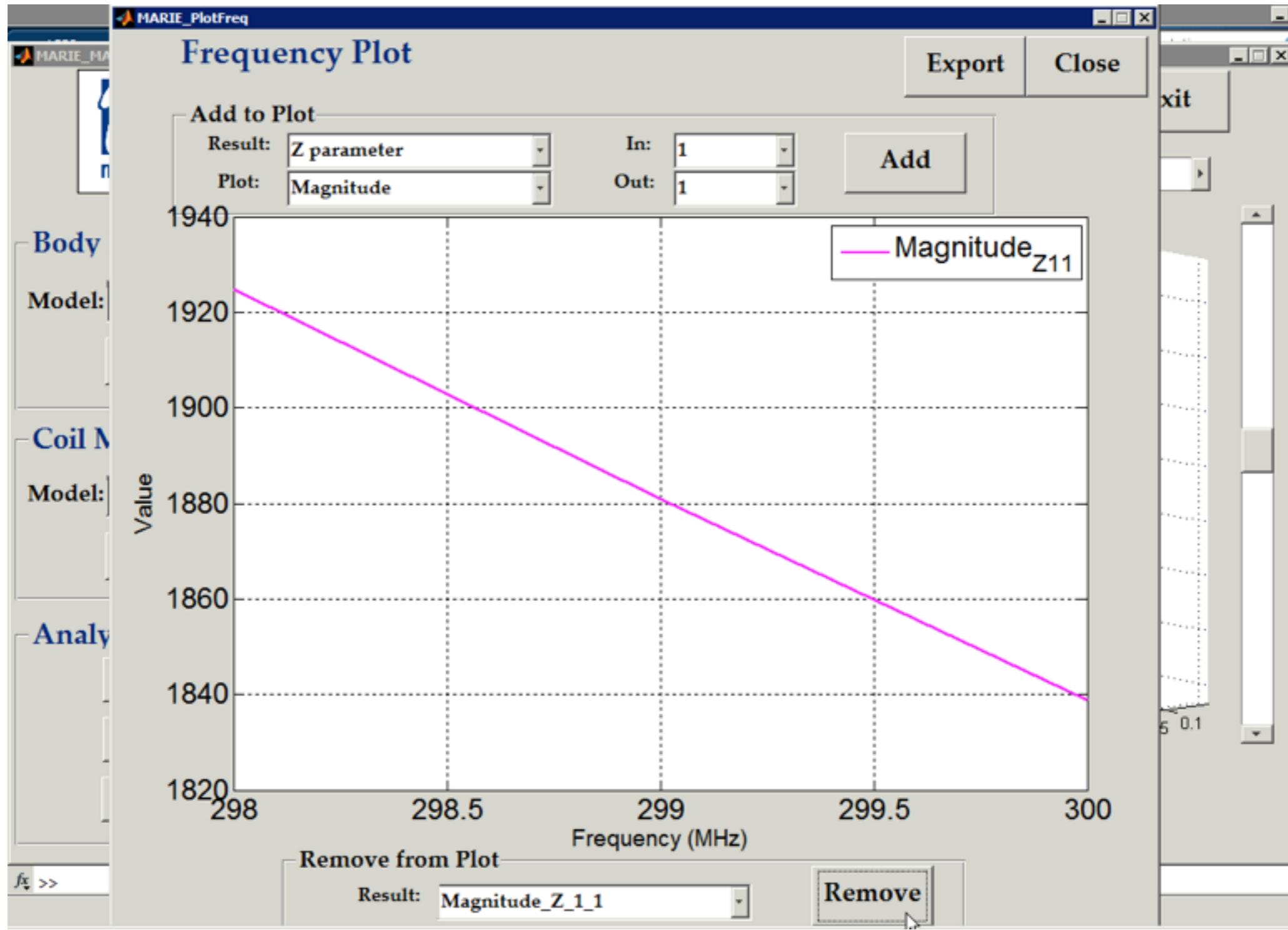


- ♦ Visualize the results

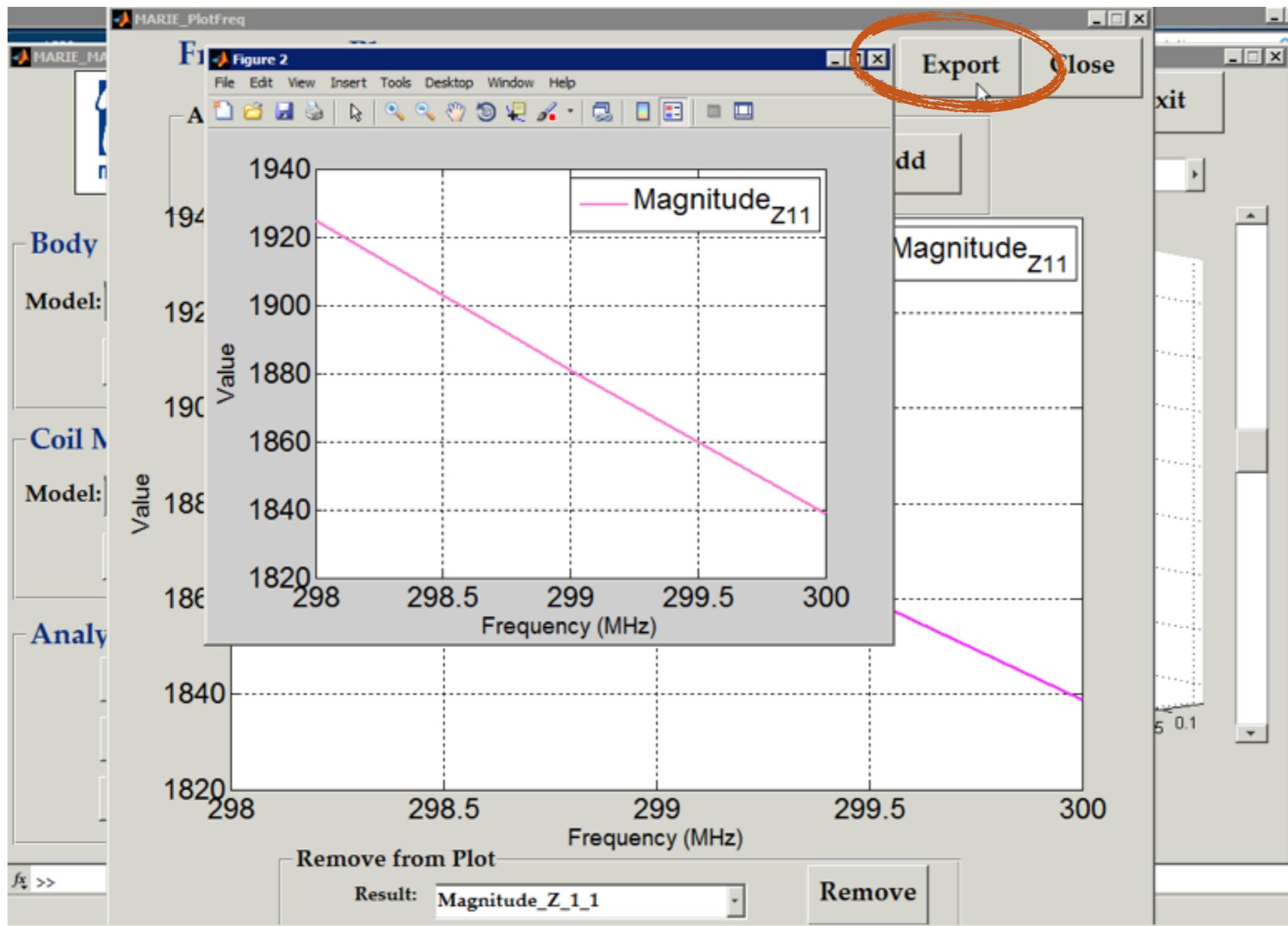
- ♦ select curve to be removed from the plot and push “remove”



- ◆ Visualize the results
  - ◆ plot will autoscale to remaining curves

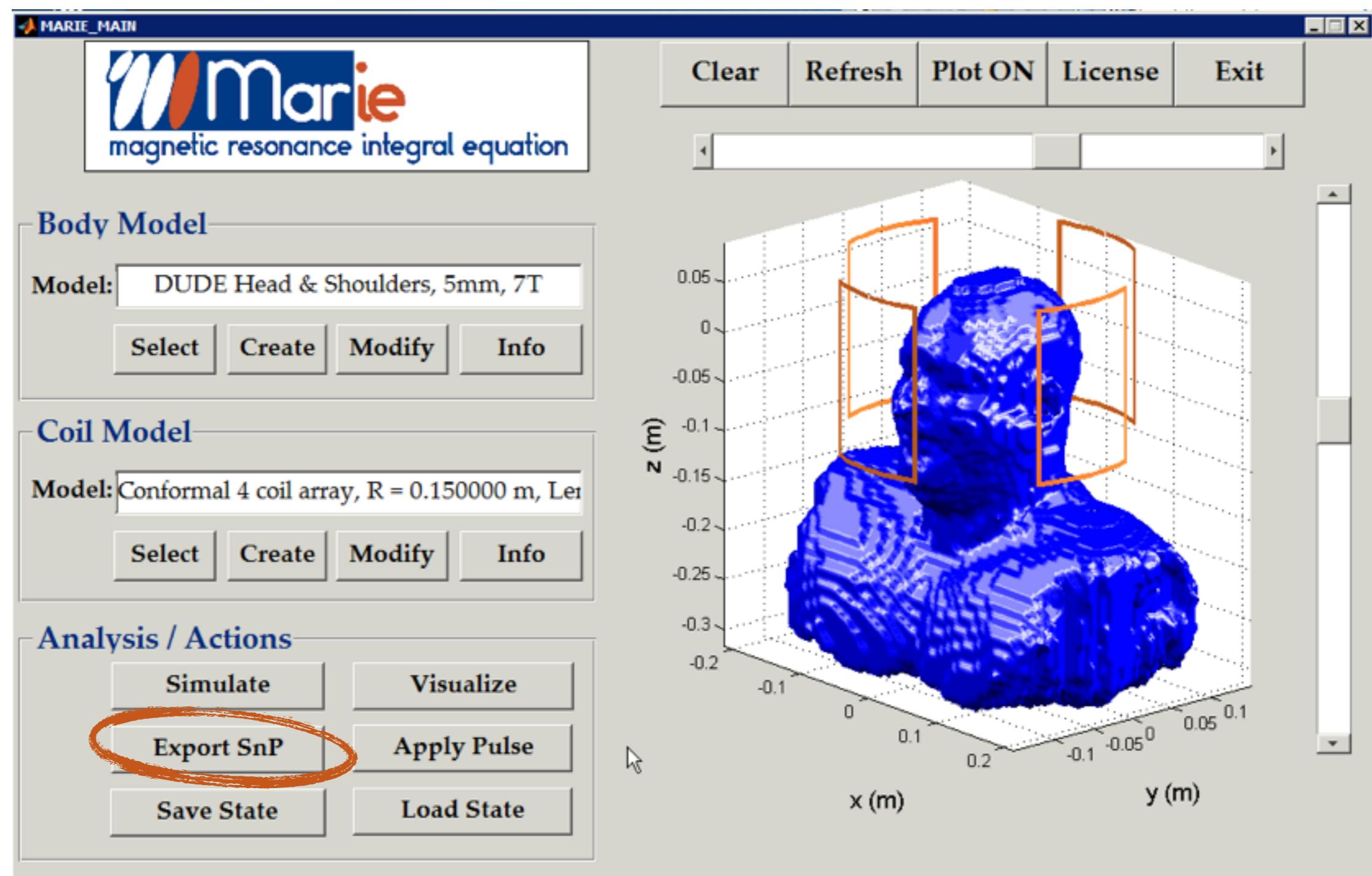


- ◆ Visualize the results
  - ◆ push “export” to generate a Matlab figure



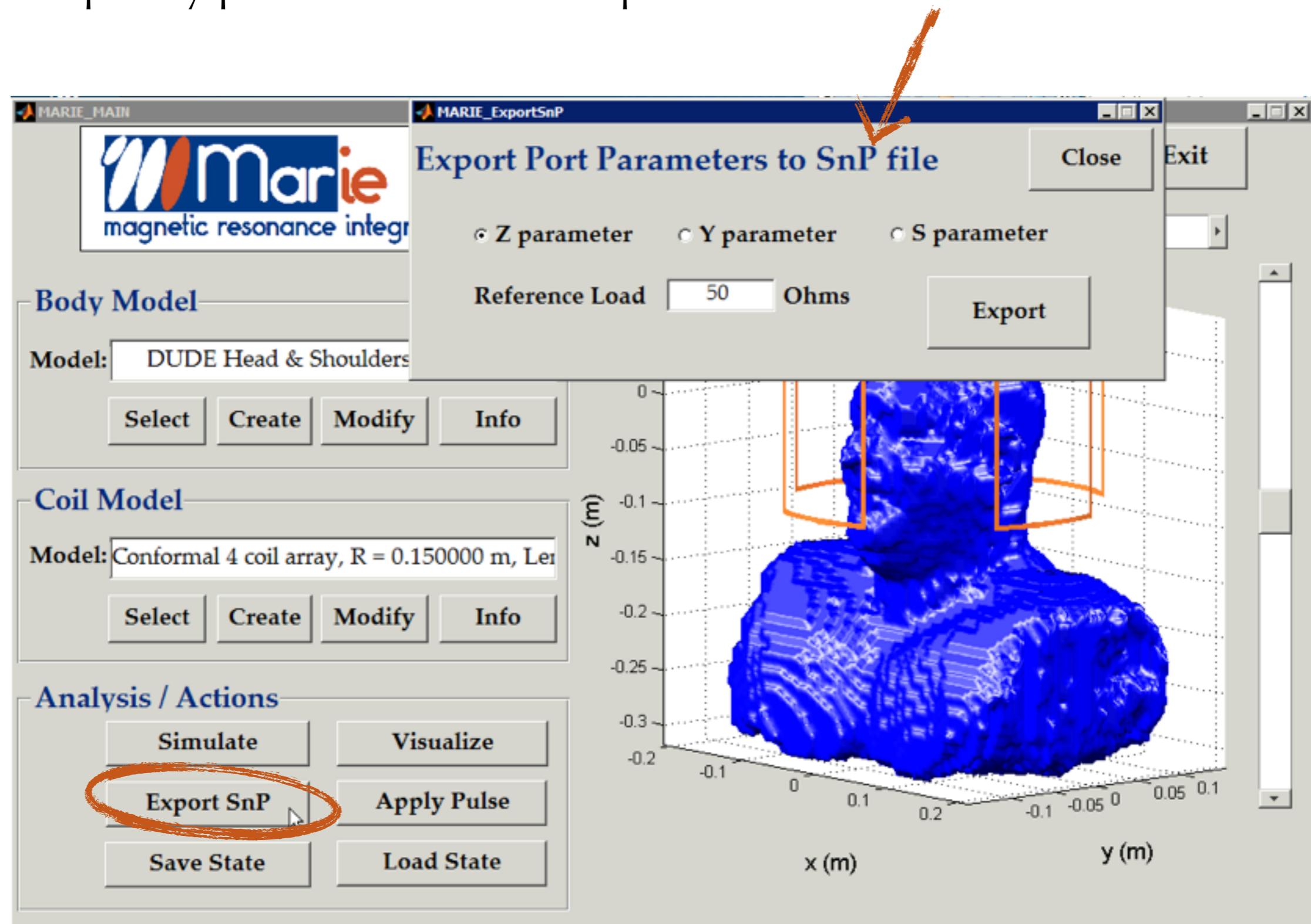
- ♦ Export the SnP parameters

- ♦ Frequency parameters can be exported to a SnP format file

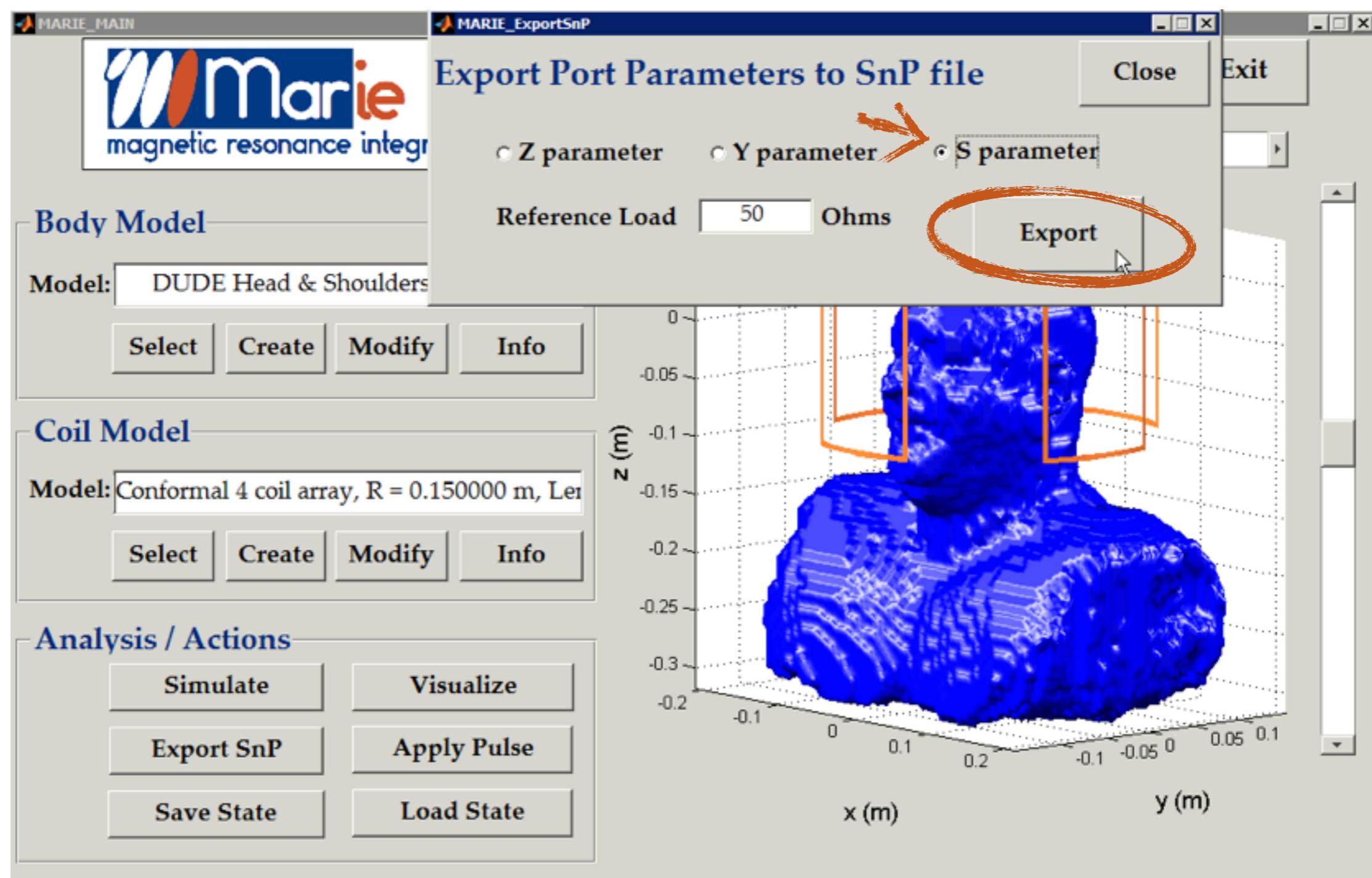


- ♦ Export the SnP parameters

- ♦ Frequency parameters can be exported to a SnP format file

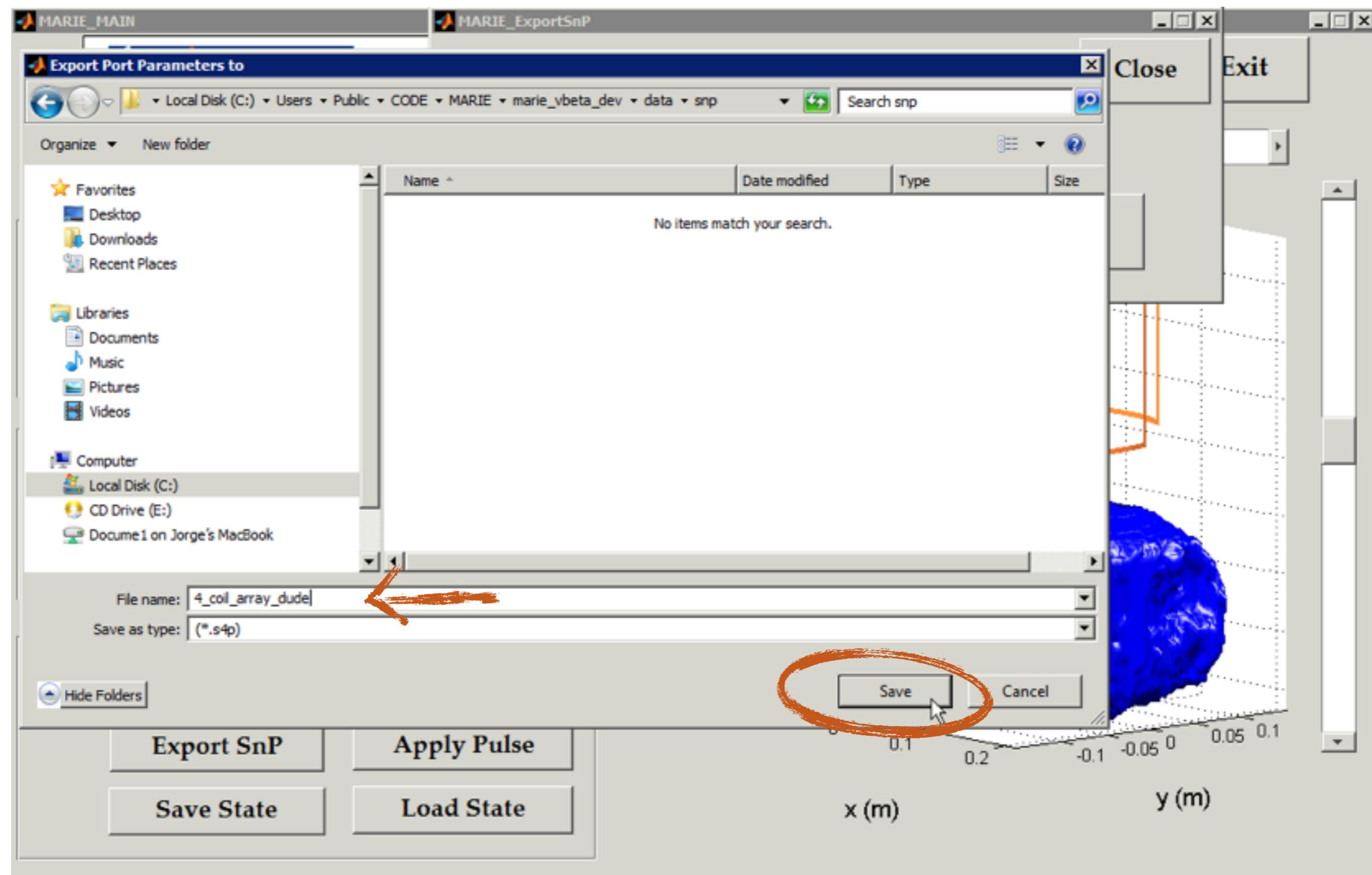


- ◆ Export the SnP parameters
  - ◆ Select the parameter to export, reference load and press export

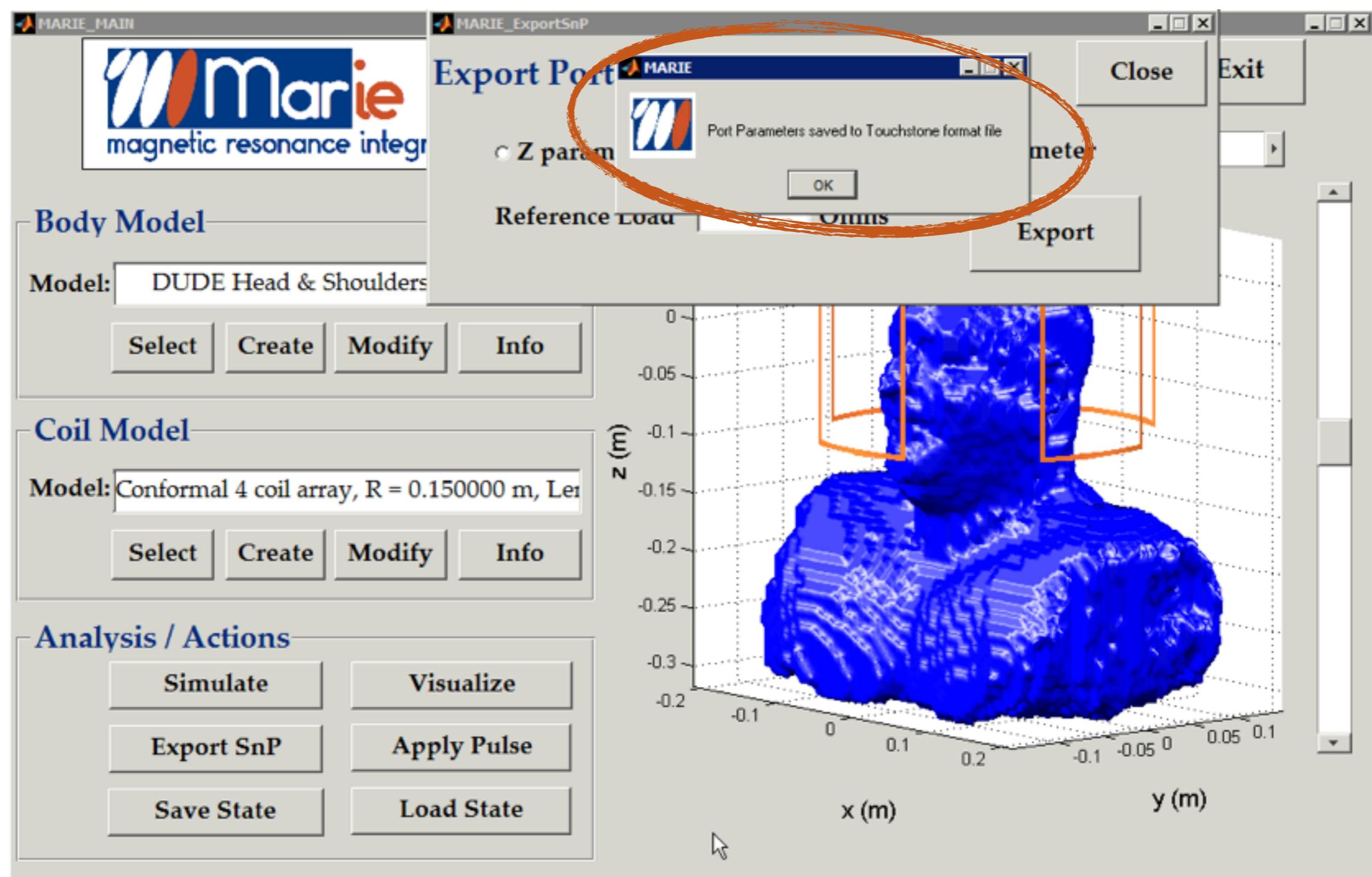


## ◆ Export the SnP parameters

- ◆ will open a window saving dialogue: select name and folder
- ◆ press save

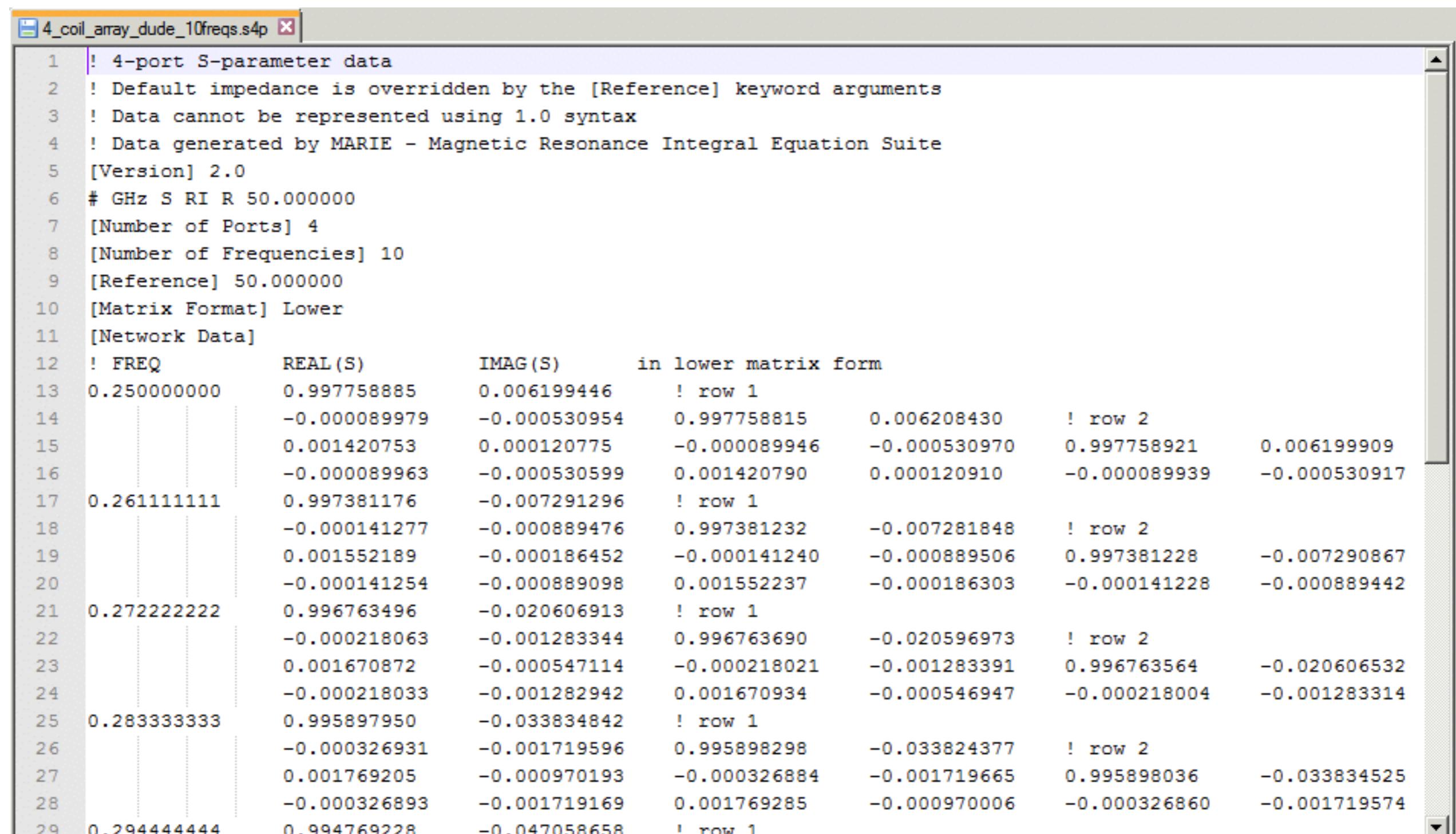


- ◆ Export the SnP parameters
  - ◆ a confirmation window will appear



## ◆ Export the SnP parameters

- ◆ the format is standard Touchstone
- ◆ can be used to interface with circuit simulators

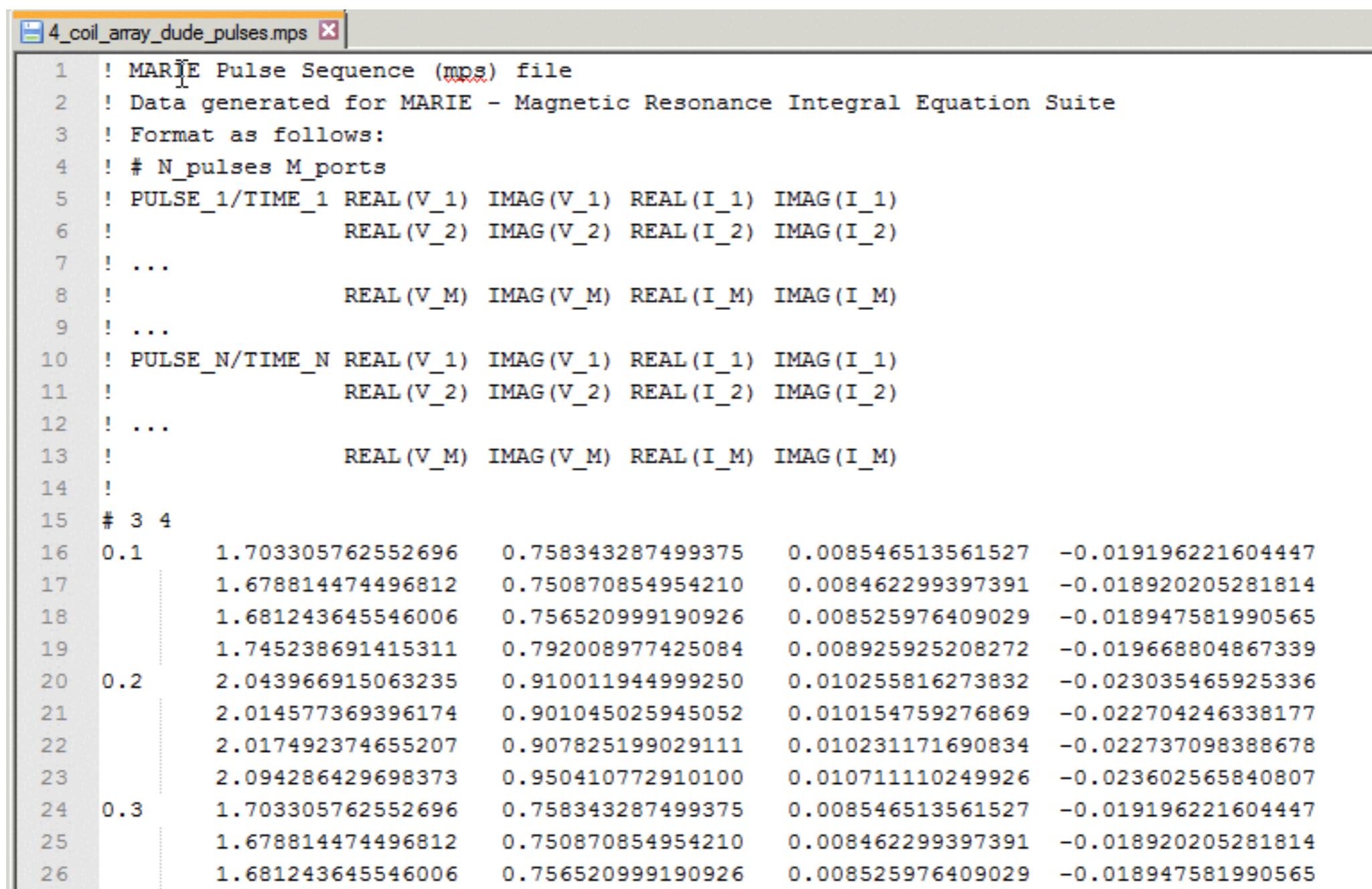


The screenshot shows a text editor window with the title bar "4\_coil\_array\_dude\_10freqs.s4p". The content of the file is a standard Touchstone S-parameter representation. It starts with header information about the number of ports (4), frequencies (10), and reference impedance (50.000000). The matrix is presented in lower triangular form, with each row labeled by its frequency value.

```
1 ! 4-port S-parameter data
2 ! Default impedance is overridden by the [Reference] keyword arguments
3 ! Data cannot be represented using 1.0 syntax
4 ! Data generated by MARIE - Magnetic Resonance Integral Equation Suite
5 [Version] 2.0
6 # GHz S RI R 50.000000
7 [Number of Ports] 4
8 [Number of Frequencies] 10
9 [Reference] 50.000000
10 [Matrix Format] Lower
11 [Network Data]
12 ! FREQ      REAL(S)      IMAG(S)      in lower matrix form
13 0.250000000  0.997758885  0.006199446  ! row 1
14          -0.000089979  -0.000530954  0.997758815  0.006208430  ! row 2
15          0.001420753  0.000120775  -0.000089946  -0.000530970  0.997758921  0.006199909
16          -0.000089963  -0.000530599  0.001420790  0.000120910  -0.000089939  -0.000530917
17 0.261111111  0.997381176  -0.007291296  ! row 1
18          -0.000141277  -0.000889476  0.997381232  -0.007281848  ! row 2
19          0.001552189  -0.000186452  -0.000141240  -0.000889506  0.997381228  -0.007290867
20          -0.000141254  -0.000889098  0.001552237  -0.000186303  -0.000141228  -0.000889442
21 0.272222222  0.996763496  -0.020606913  ! row 1
22          -0.000218063  -0.001283344  0.996763690  -0.020596973  ! row 2
23          0.001670872  -0.000547114  -0.000218021  -0.001283391  0.996763564  -0.020606532
24          -0.000218033  -0.001282942  0.001670934  -0.000546947  -0.000218004  -0.001283314
25 0.283333333  0.995897950  -0.033834842  ! row 1
26          -0.000326931  -0.001719596  0.995898298  -0.033824377  ! row 2
27          0.001769205  -0.000970193  -0.000326884  -0.001719665  0.995898036  -0.033834525
28          -0.000326893  -0.001719169  0.001769285  -0.000970006  -0.000326860  -0.001719574
29 0.294444444  0.994769228  -0.047058658  ! row 1
```

## ◆ Apply Pulse

- ◆ The SnP can be used in a circuit co-simulation environment
  - ◆ tuning and matching the coils
  - ◆ a pulse will generate voltage/currents at the ports



```
4_coil_array_dude_pulses.mps X
1 ! MARIE Pulse Sequence (mps) file
2 ! Data generated for MARIE - Magnetic Resonance Integral Equation Suite
3 ! Format as follows:
4 ! # N_pulses M_ports
5 ! PULSE_1/TIME_1 REAL(V_1) IMAG(V_1) REAL(I_1) IMAG(I_1)
6 !             REAL(V_2) IMAG(V_2) REAL(I_2) IMAG(I_2)
7 !
8 ! ...
9 !             REAL(V_M) IMAG(V_M) REAL(I_M) IMAG(I_M)
10 !
11 ! ...
12 !
13 !             REAL(V_M) IMAG(V_M) REAL(I_M) IMAG(I_M)
14 !
15 # 3 4
16 0.1    1.703305762552696   0.758343287499375   0.008546513561527   -0.019196221604447
17      1.678814474496812   0.750870854954210   0.008462299397391   -0.018920205281814
18      1.681243645546006   0.756520999190926   0.008525976409029   -0.018947581990565
19      1.745238691415311   0.792008977425084   0.008925925208272   -0.019668804867339
20 0.2    2.043966915063235   0.910011944999250   0.010255816273832   -0.023035465925336
21      2.014577369396174   0.901045025945052   0.010154759276869   -0.022704246338177
22      2.017492374655207   0.907825199029111   0.010231171690834   -0.022737098388678
23      2.094286429698373   0.950410772910100   0.010711110249926   -0.023602565840807
24 0.3    1.703305762552696   0.758343287499375   0.008546513561527   -0.019196221604447
25      1.678814474496812   0.750870854954210   0.008462299397391   -0.018920205281814
26      1.681243645546006   0.756520999190926   0.008525976409029   -0.018947581990565
```

## ♦ Apply Pulse

- ♦ .mps (marie pulse sequence) files encapsulate this info
  - ♦ a number N of pulses, with M ports pairs of complex voltage / current
  - ♦ each pulse can be defined by a number, or a time step

# pulses, ports

pulse time step

```

1 ! MARIE Pulse Sequence (.mps) file
2 ! Data generated for MARIE - Magnetic Resonance Integral Equation Suite
3 ! Format as follows:
4 ! # N_pulses M_ports
5 ! PULSE_1/TIME_1 REAL(V_1) IMAG(V_1) REAL(I_1) IMAG(I_1)
6 !           REAL(V_2) IMAG(V_2) REAL(I_2) IMAG(I_2)
7 !
8 ! ...
9 !           REAL(V_M) IMAG(V_M) REAL(I_M) IMAG(I_M)
10 !
11 !
12 ! ...
13 !
14 !
15 !           REAL(V_M) IMAG(V_M) REAL(I_M) IMAG(I_M)
16 !
17 ! # 3 4
18 !
19 !
20 ! 0.1
21 !     1.703305762552696 0.758343287499375 0.008546513561527 -0.019196221604447
22 !     1.678814474496812 0.750870854954210 0.008462299397391 -0.018920205281814
23 !     1.681243645546006 0.756520999190926 0.008525976409029 -0.018947581990565
24 !     1.745238691415311 0.792008977425084 0.008925925208272 -0.019668804867339
25 ! 0.2
26 !     2.043966915063235 0.910011944999250 0.010255816273832 -0.023035465925336
27 !     2.014577369396174 0.901045025945052 0.010154759276869 -0.022704246338177
28 !     2.017492374655207 0.907825199029111 0.010231171690834 -0.022737098388678
29 !     2.094286429698373 0.950410772910100 0.010711110249926 -0.023602565840807
30 !
31 ! 0.3
32 !     1.703305762552696 0.758343287499375 0.008546513561527 -0.019196221604447
33 !     1.678814474496812 0.750870854954210 0.008462299397391 -0.018920205281814
34 !     1.681243645546006 0.756520999190926 0.008525976409029 -0.018947581990565

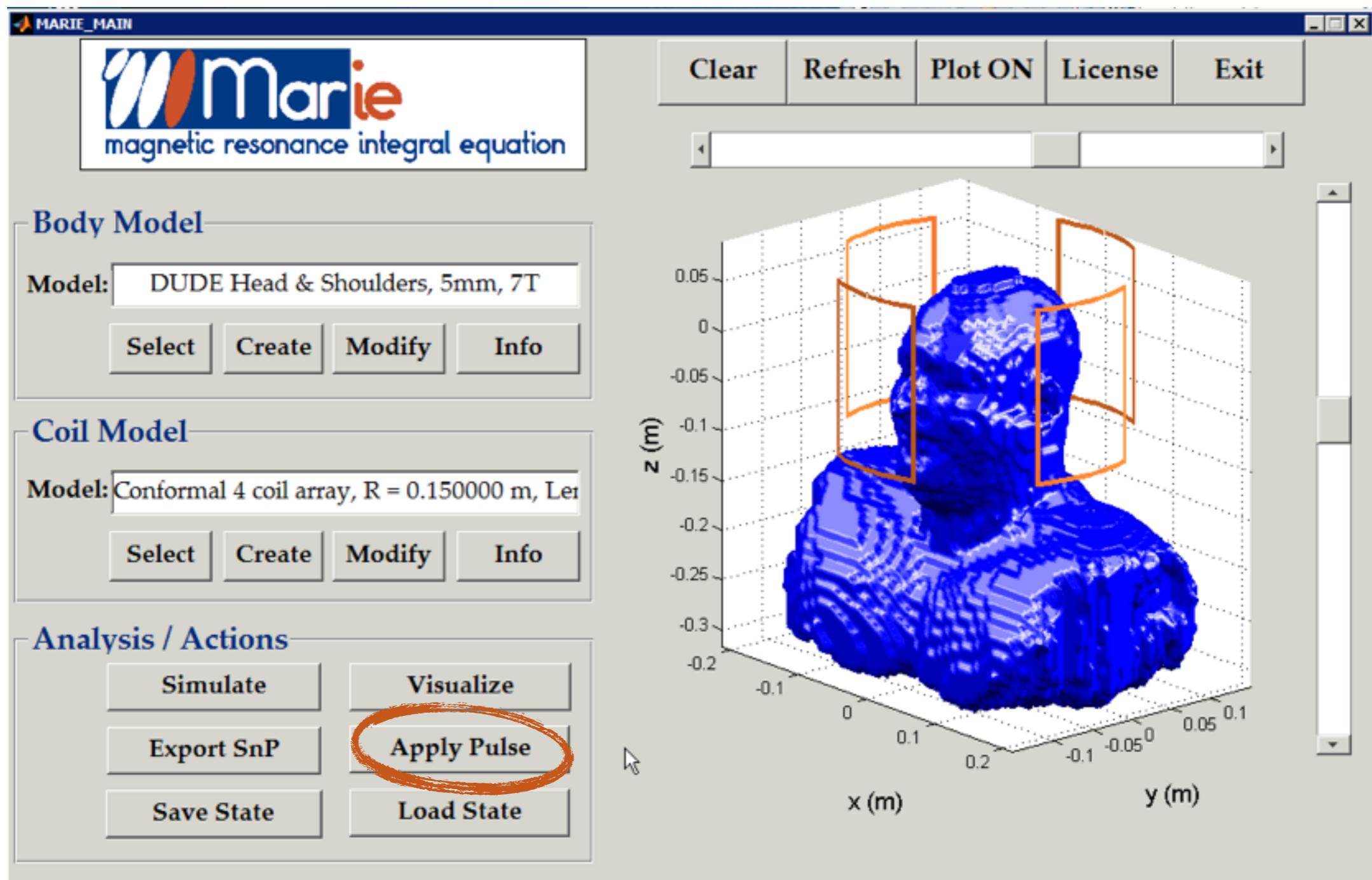
```

real(V)      imag(V)      real(I)      imag(I)

pulse 0.1

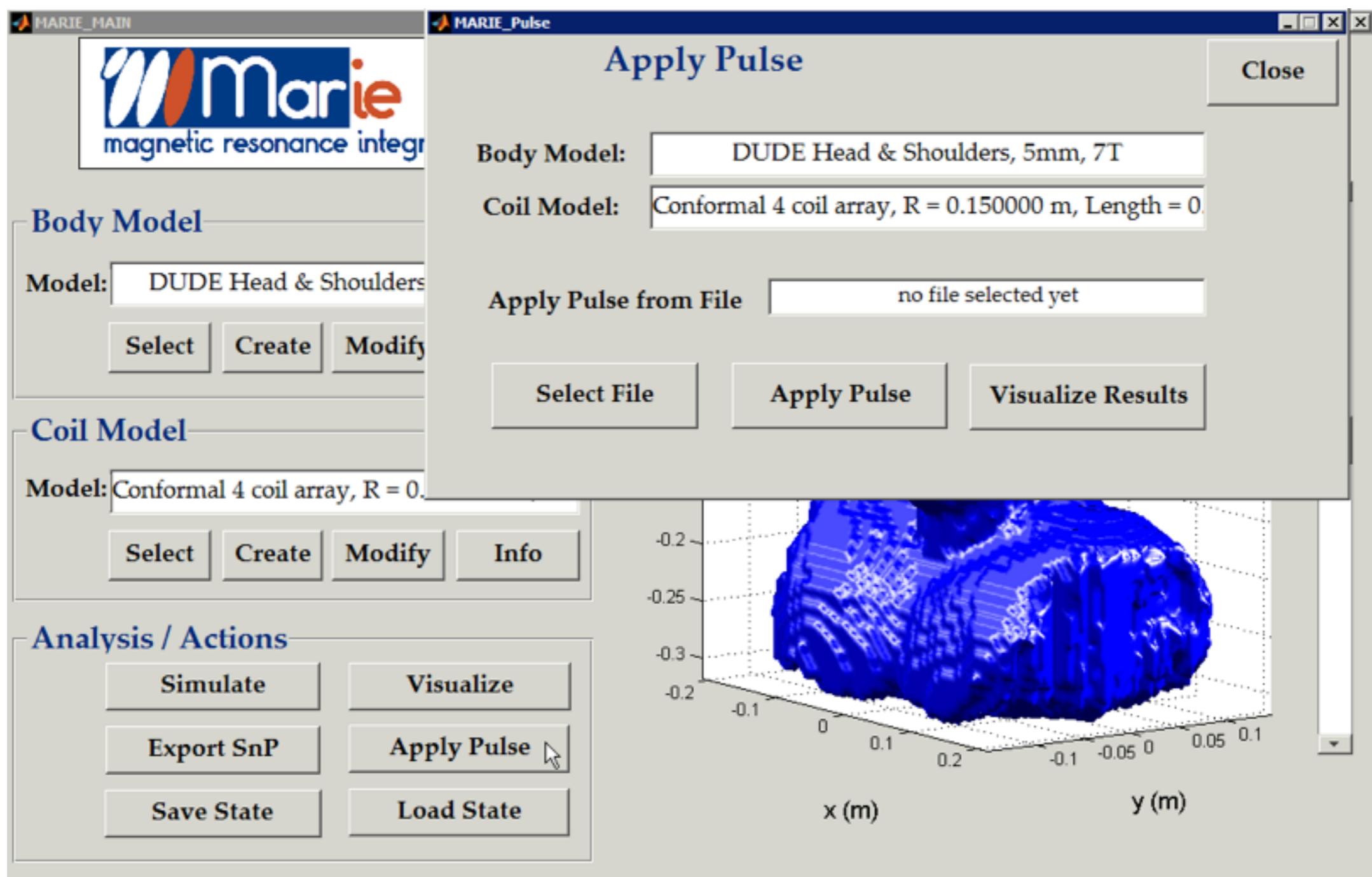
pulse 0.2

- ◆ Apply Pulse
  - ◆ push “Apply Pulse” to load a .mps file



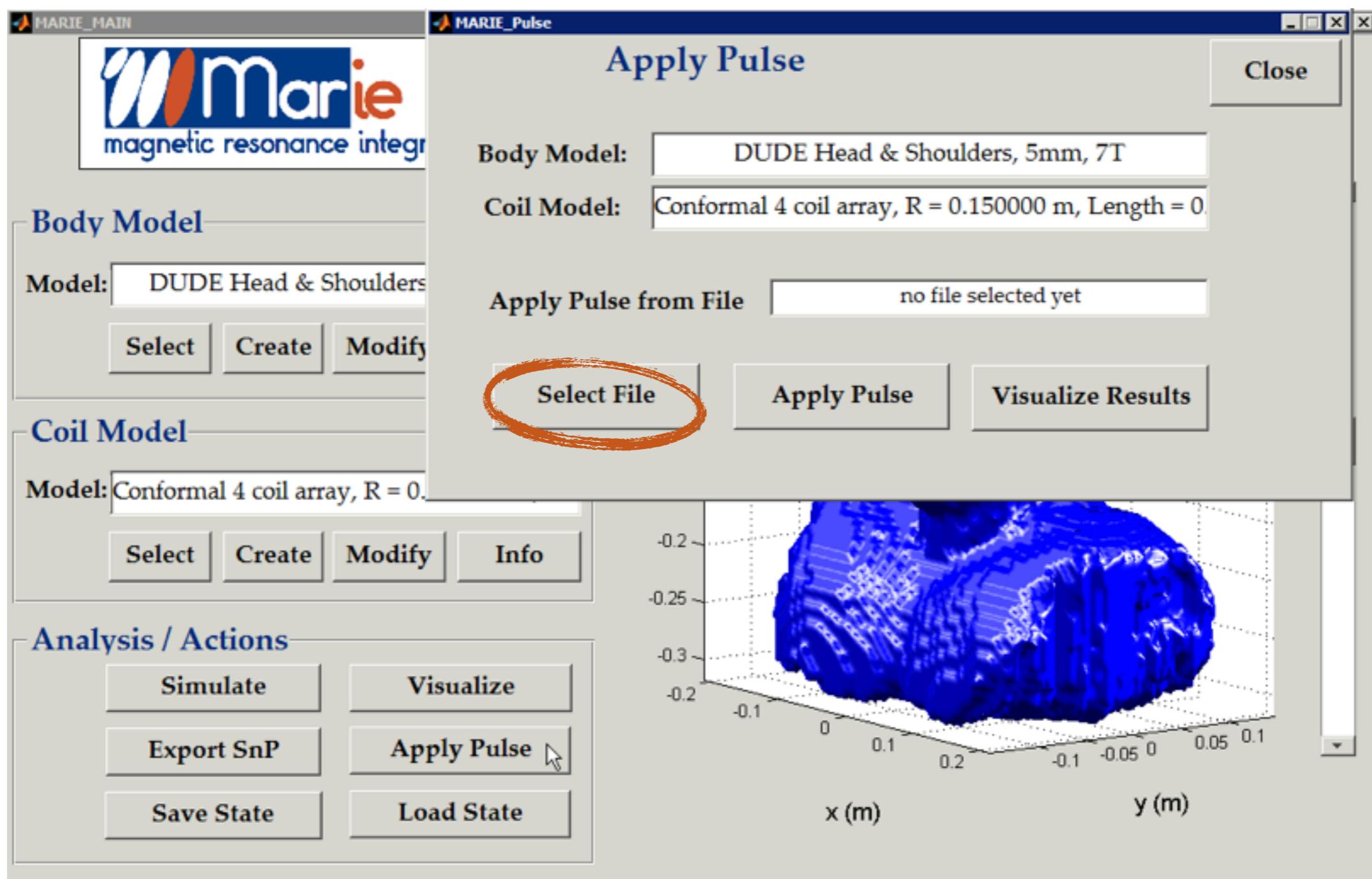
## ♦ Apply Pulse

- ♦ push “Apply Pulse” to load a .mps file
- ♦ opens the dialogue box



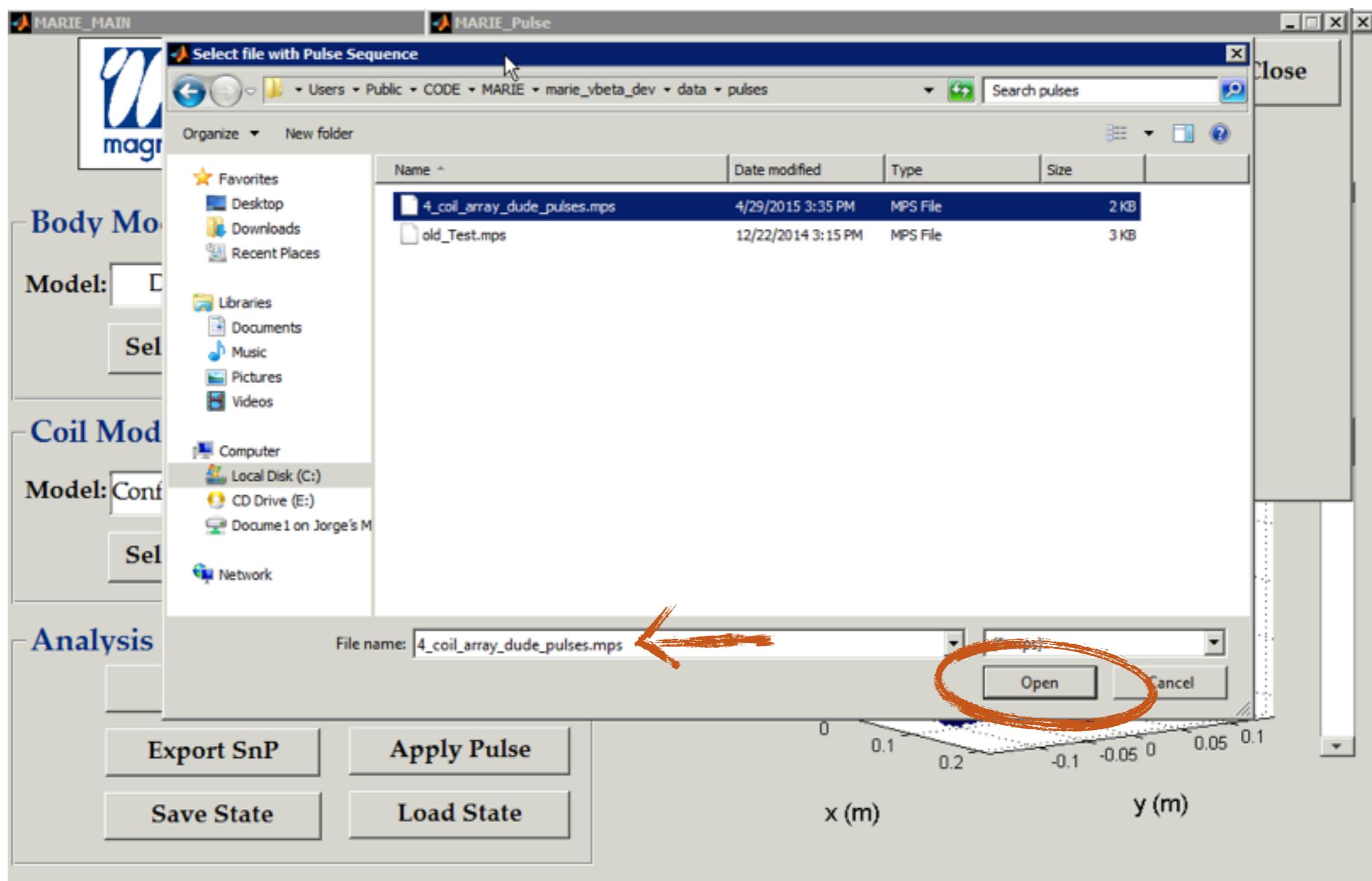
## ♦ Apply Pulse

- ♦ select the file: will open a navigation window



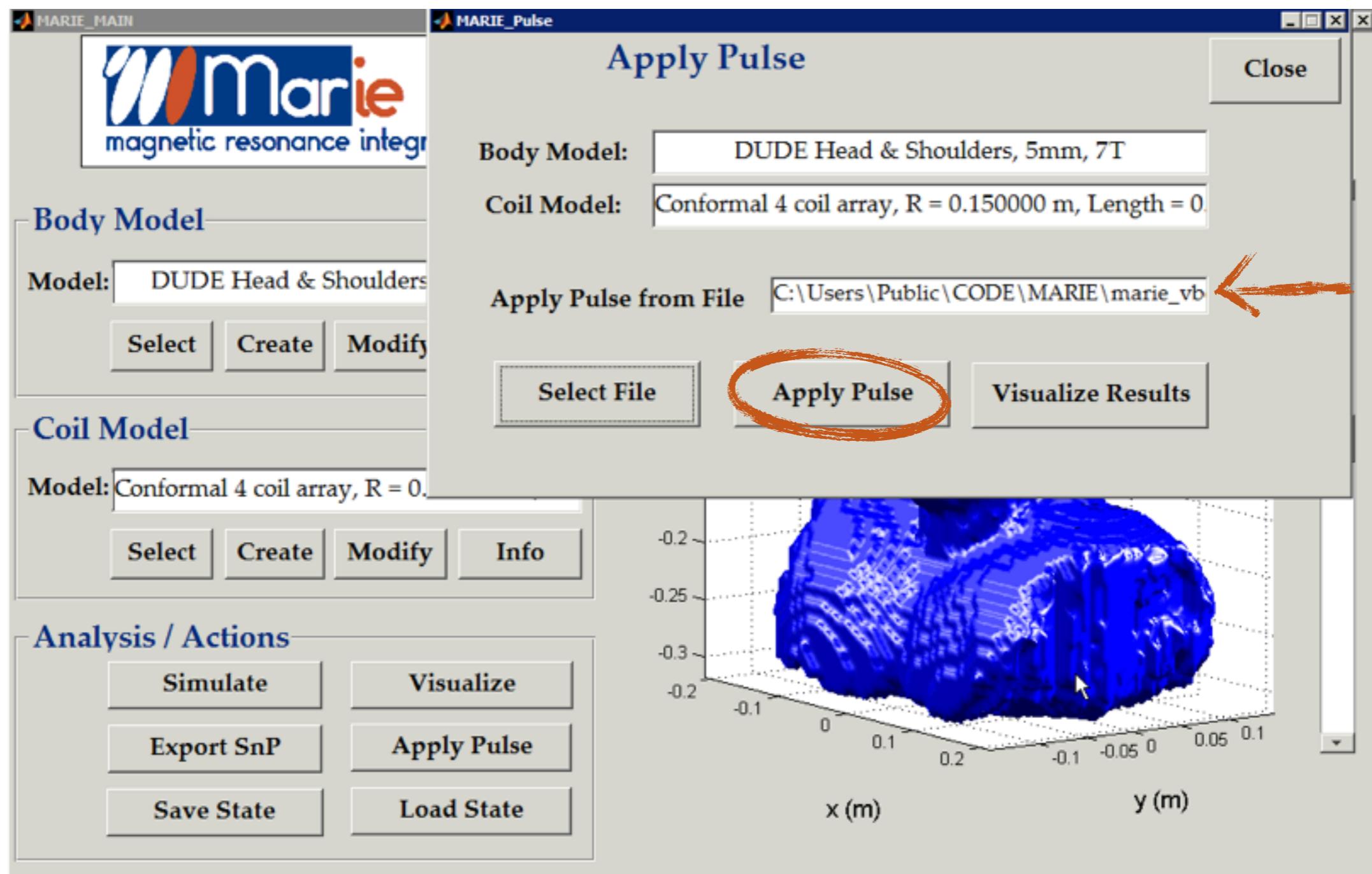
## ◆ Apply Pulse

- ◆ select the file: will open a navigation window
- ◆ selects the .mps file, and open it



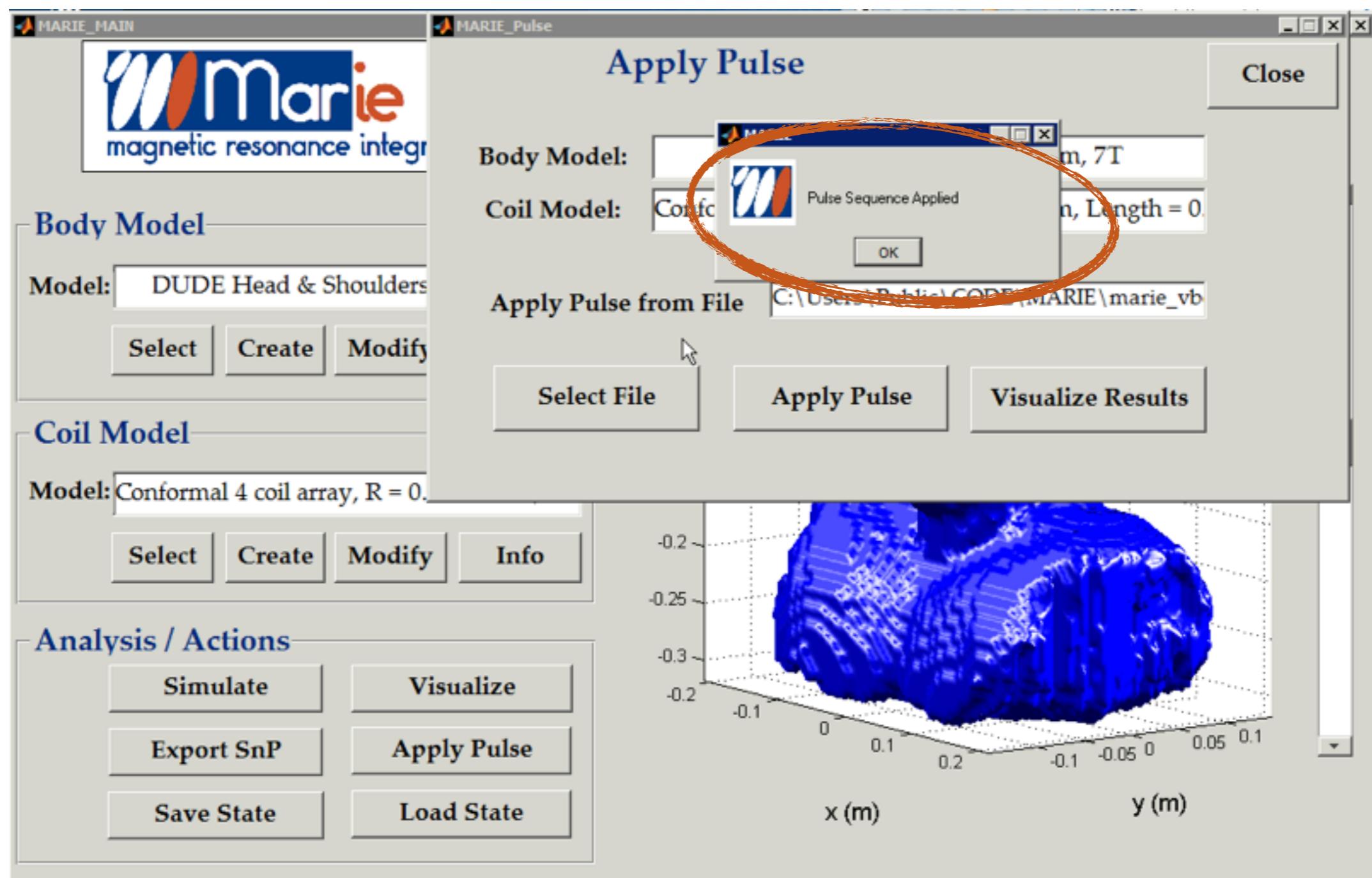
## ♦ Apply Pulse

- ◆ file will be selected as source
- ◆ push apply pulse



## ♦ Apply Pulse

- ♦ a window will confirm



## ◆ Apply Pulse

- ◆ and info in the command line

Command Window

```

-----
Applying pulse sequence from:
C:\Users\Public\CODE\MARIE\marie_vbeta_dev\data\pulses\4_coil_array_dude_pulses.mps

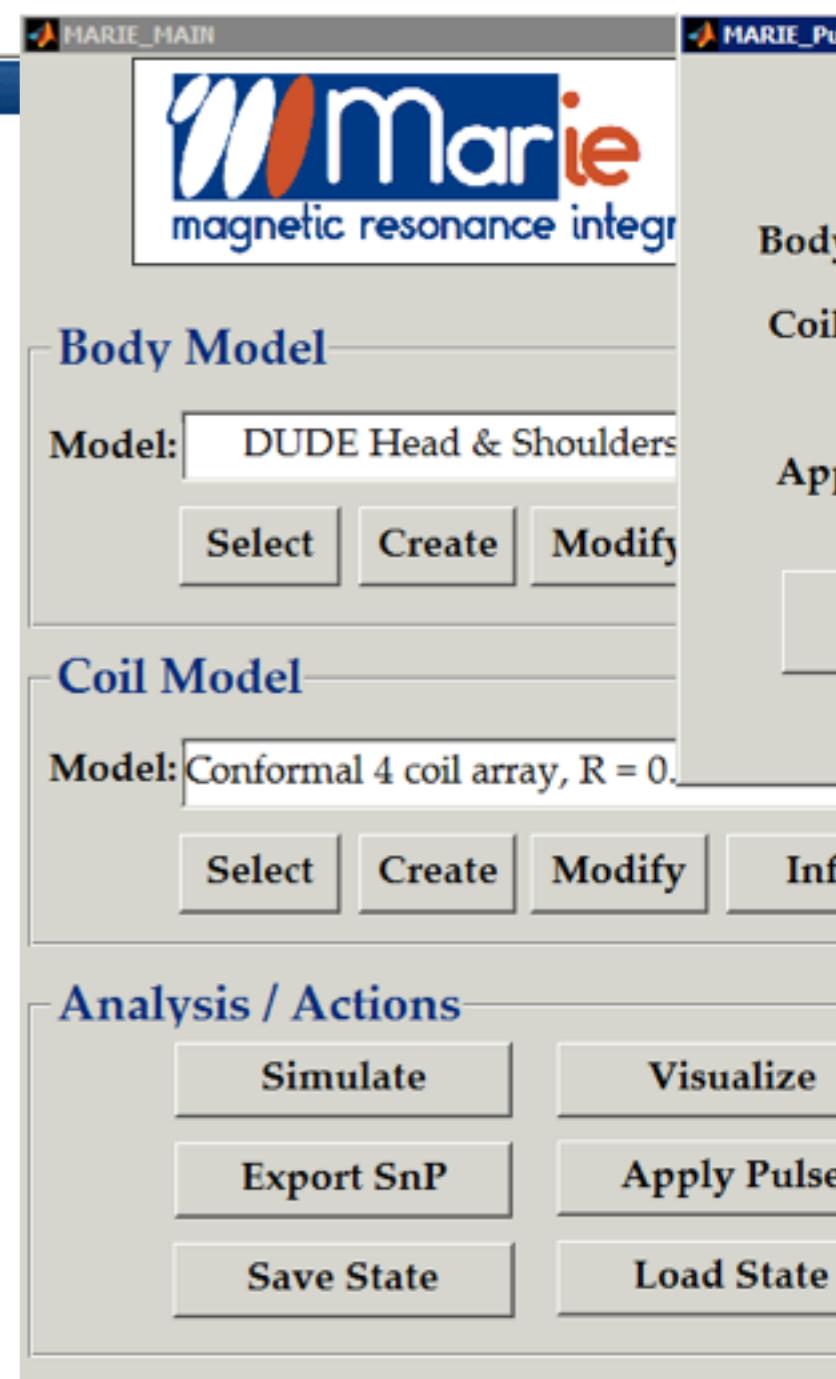
Pulse sequence applied, overall time 9.28 [sec]
-----

```

```

>> whos
  Name      Size            Bytes  Class        Attributes
  COIL      1x1             1039194  struct    global
  FIGIDX    1x1              8     double    global
  PULSE     1x1            1573581880  struct    global
  RHBM      1x1            53468988   struct    global
  SOL       1x1            1993206504  struct    global
  p          1x6341           12682   char

```



## ♦ Apply Pulse

- ♦ and info in the command line
- ♦ PULSE struct will be filled with data: pulses instead of channels

**Command Window**

```
Pulse sequence applied, overall time 9.28 [sec]
-----
>> whos
  Name      Size            Bytes  Class    Attributes
  COIL      1x1             1039194  struct   global
  FIGIDX    1x1              8     double  global
  PULSE     1x1            1573581880  struct   global
  RHBM      1x1             53468988  struct   global
  SOL       1x1            1993206504  struct   global
  p         1x6341           12682   char

>> PULSE
PULSE =
  Jcoil: [364x3x3 double]
  Jsol: [6-D double]
  Ssol: [5-D double]
  Esol: [6-D double]
  Bsol: [6-D double]
  Pabs: [3x3 double]
  Gsar: [3x3 double]
  step: [3x1 double]
  freq: [298000000 299000000 300000000]
  Ipulse: [4x3 double]
  Vpulse: [4x3 double]
```

**MARIE\_MAIN**

**MARIE\_Pulse**

**Apply P**

**Body Model:**

**Coil Model:**  Conformal

**Apply Pulse from File**

**Select File**

**Body Model**

**Model:** DUDE Head & Shoulders

**Select** **Create** **Modify**

**Coil Model**

**Model:** Conformal 4 coil array, R = 0.

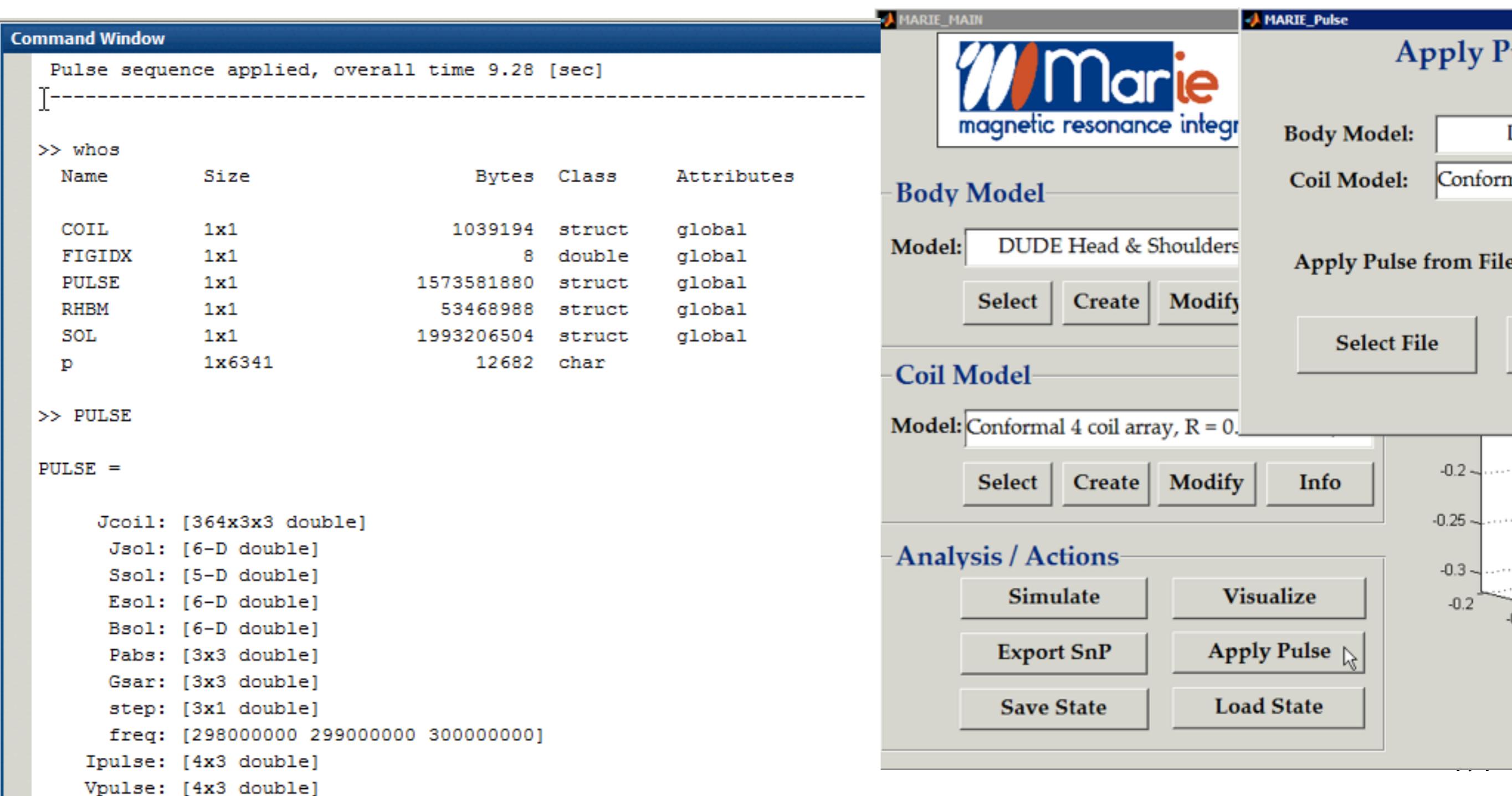
**Select** **Create** **Modify** **Info**

**Analysis / Actions**

**Simulate** **Visualize**

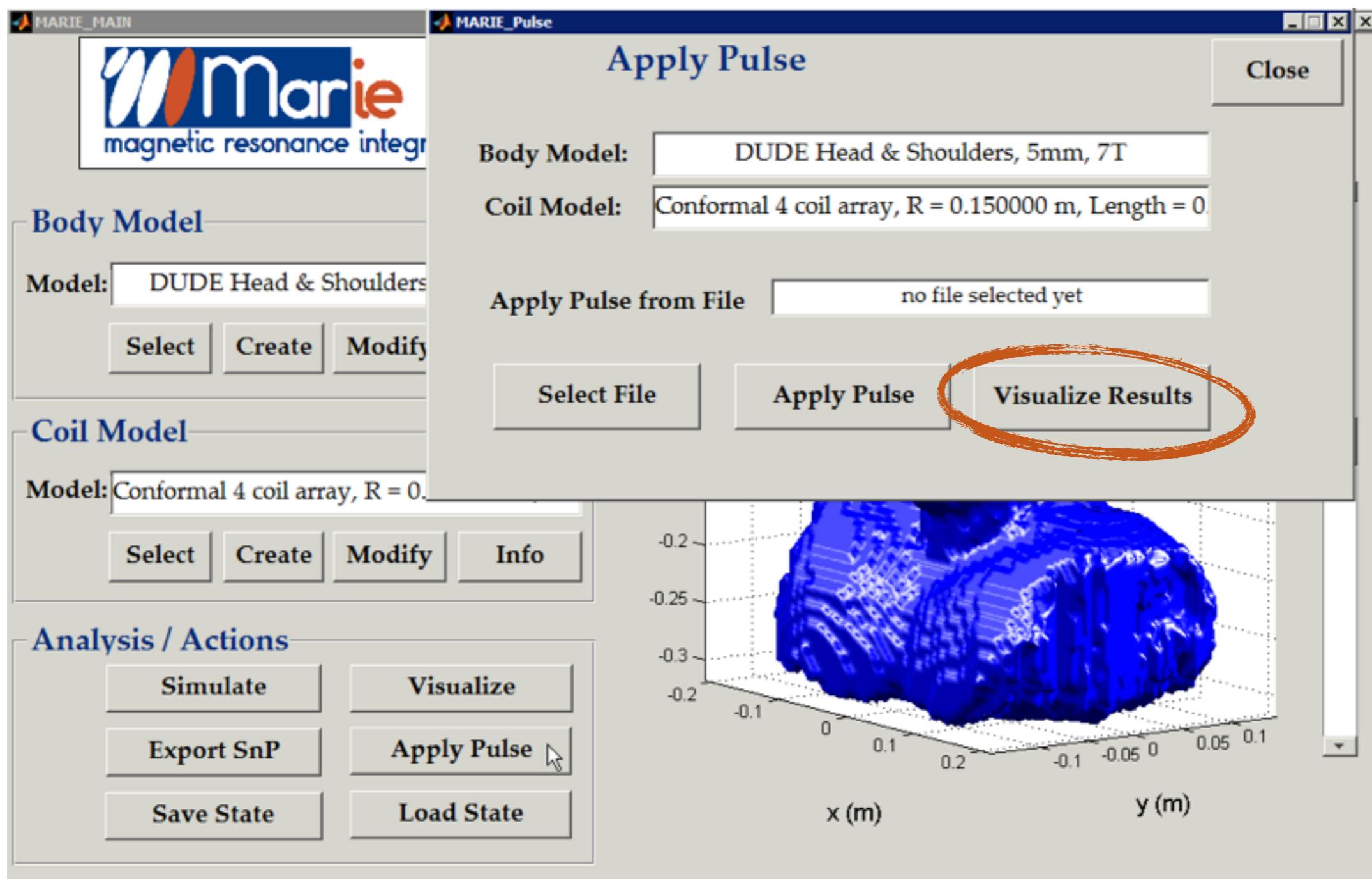
**Export SnP** **Apply Pulse**

**Save State** **Load State**



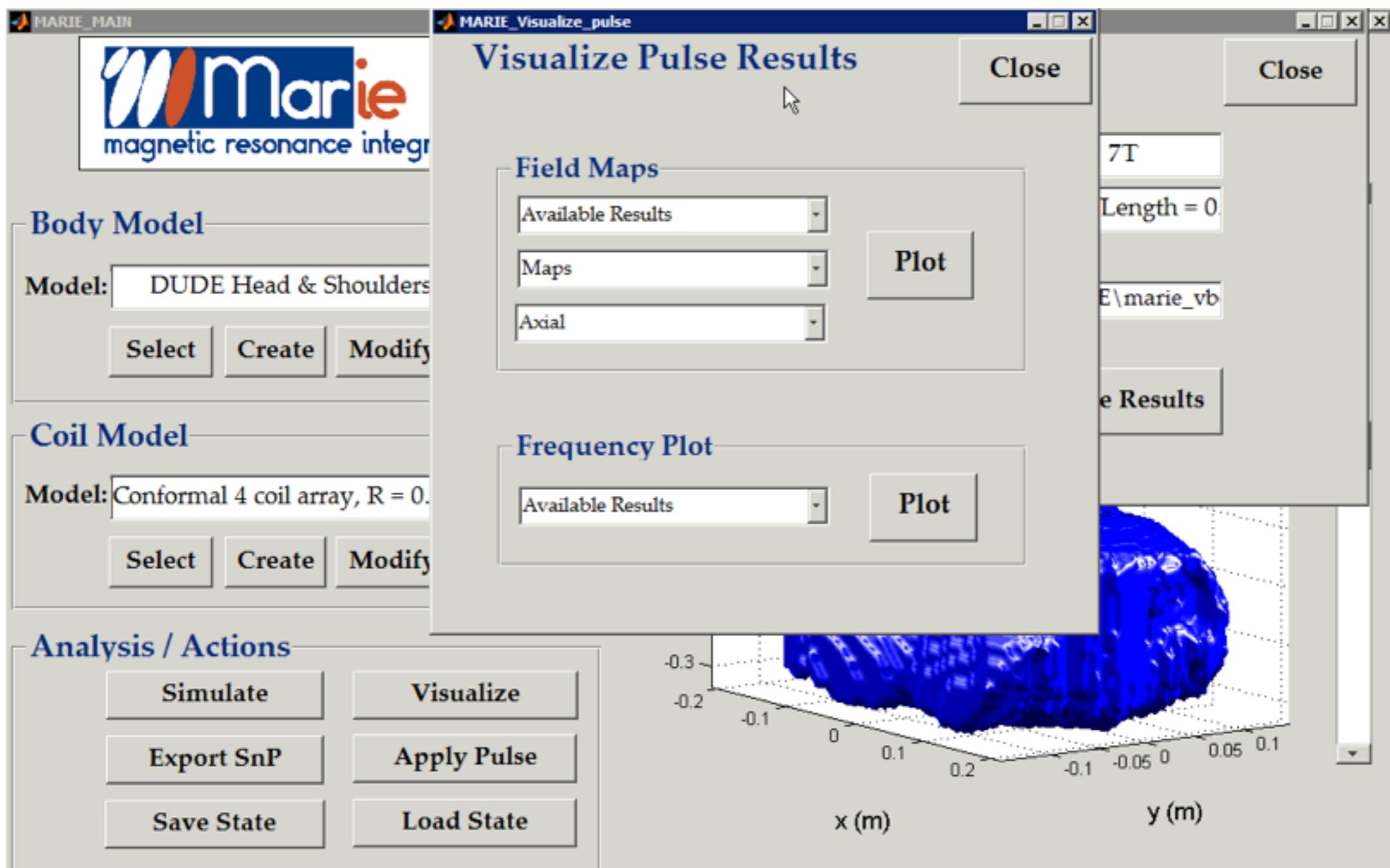
## ♦ Apply Pulse

- ♦ from the same window, push visualize to call the plot manager
- ♦ it will plot the pulse related results



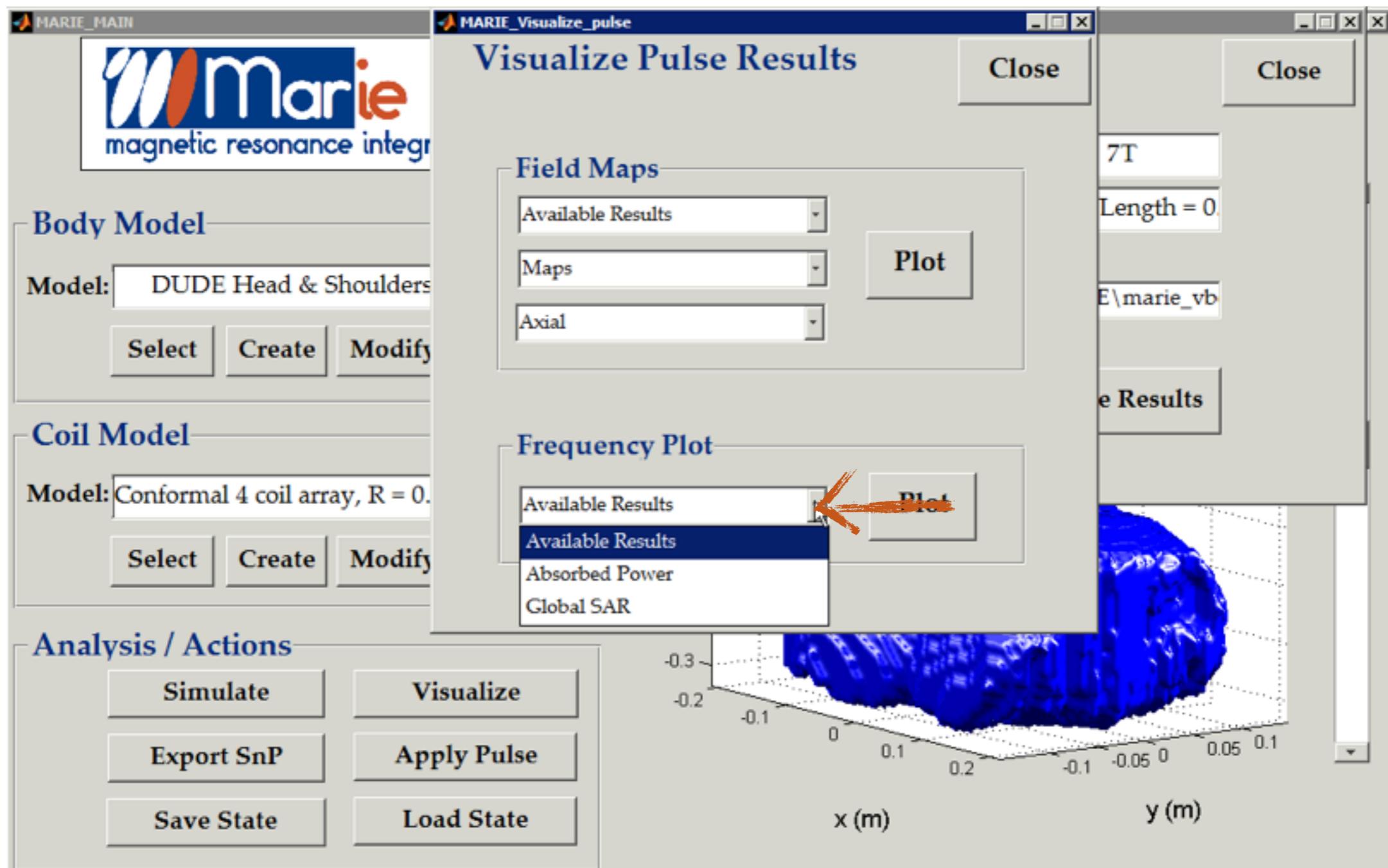
## ◆ Apply Pulse

- ◆ same interface as the main visualization window



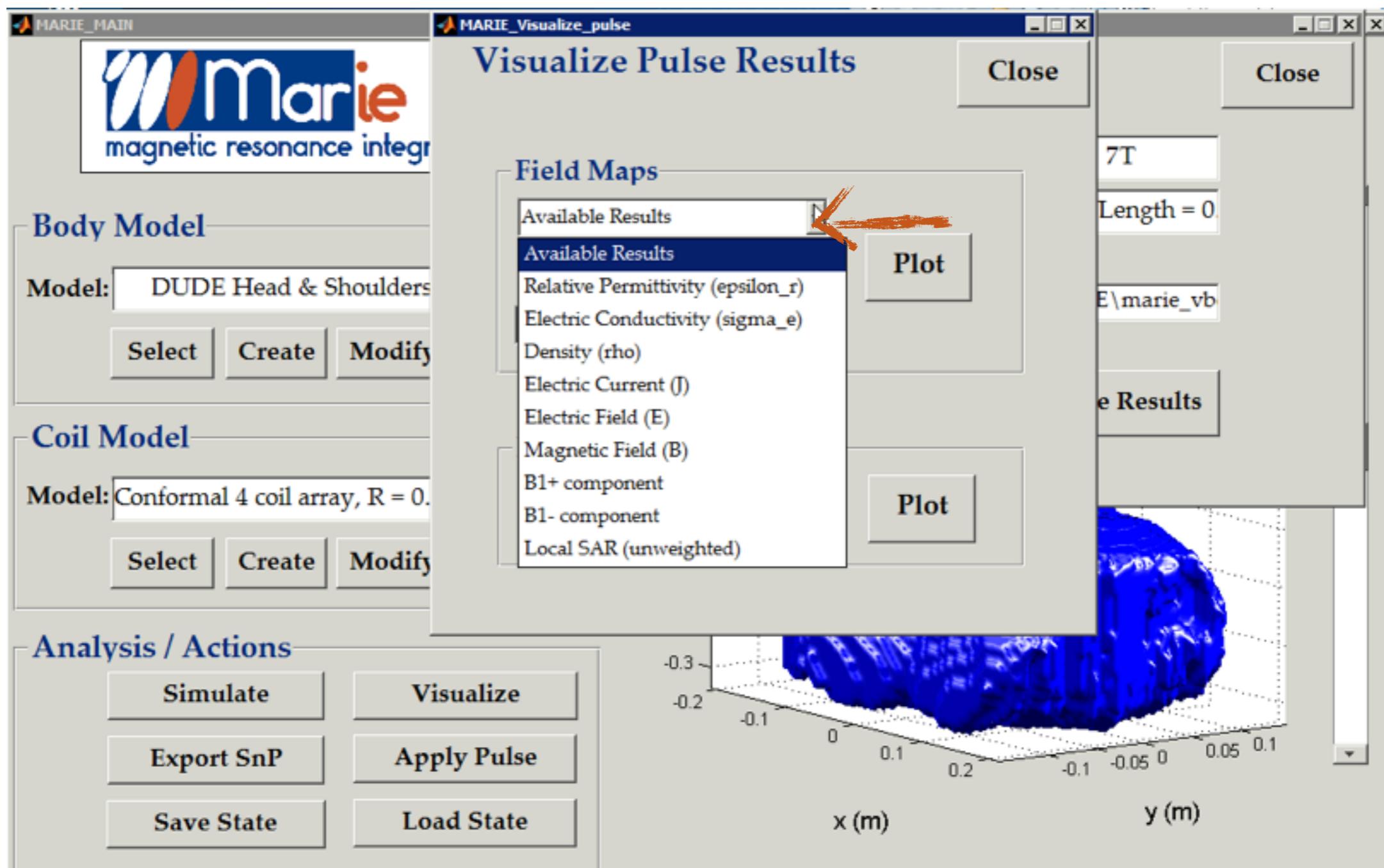
## ♦ Apply Pulse

- ♦ same interface as the main visualization window
  - ♦ frequency domain plots



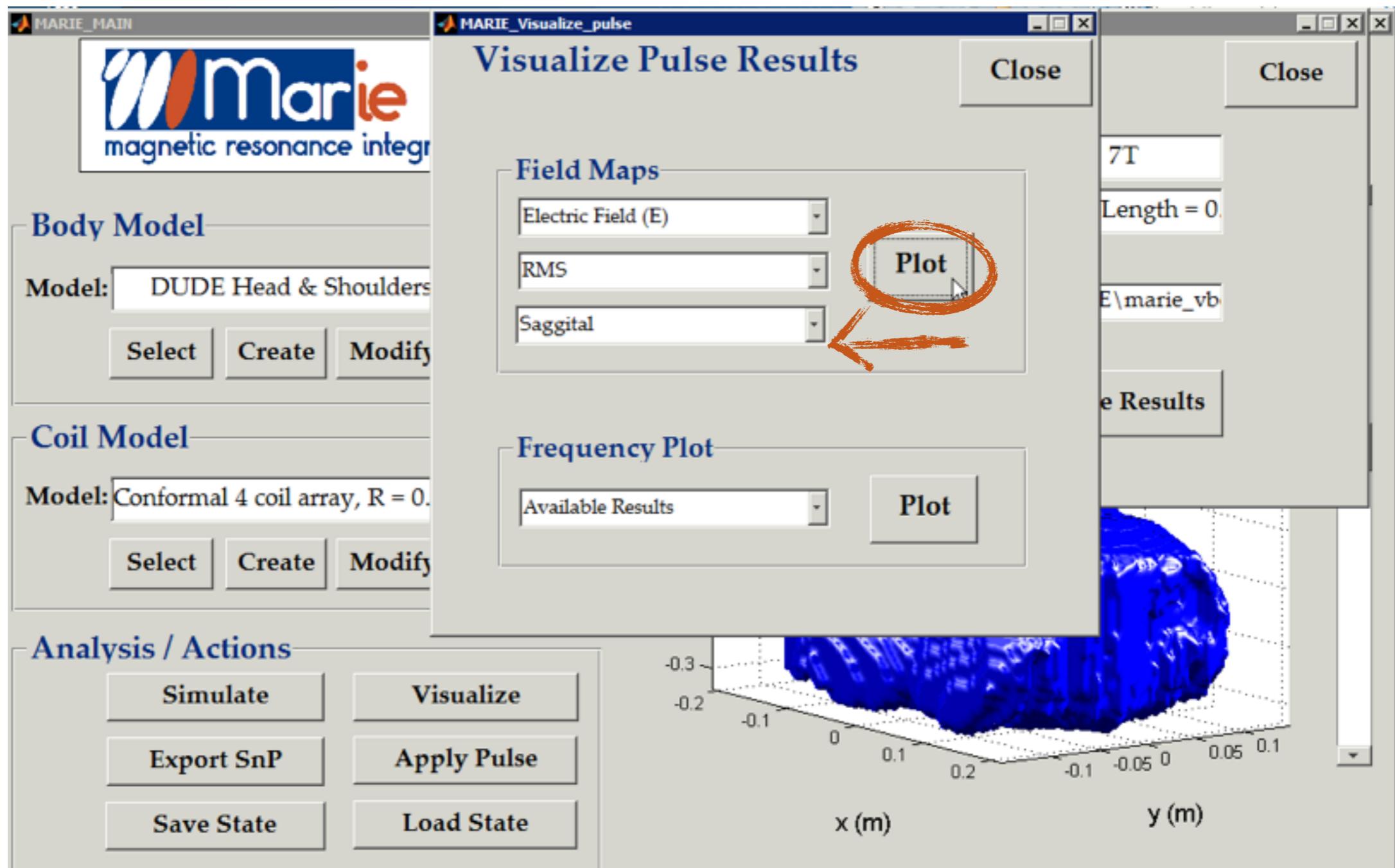
## ◆ Apply Pulse

- ◆ same interface as the main visualization window
  - ◆ and maps



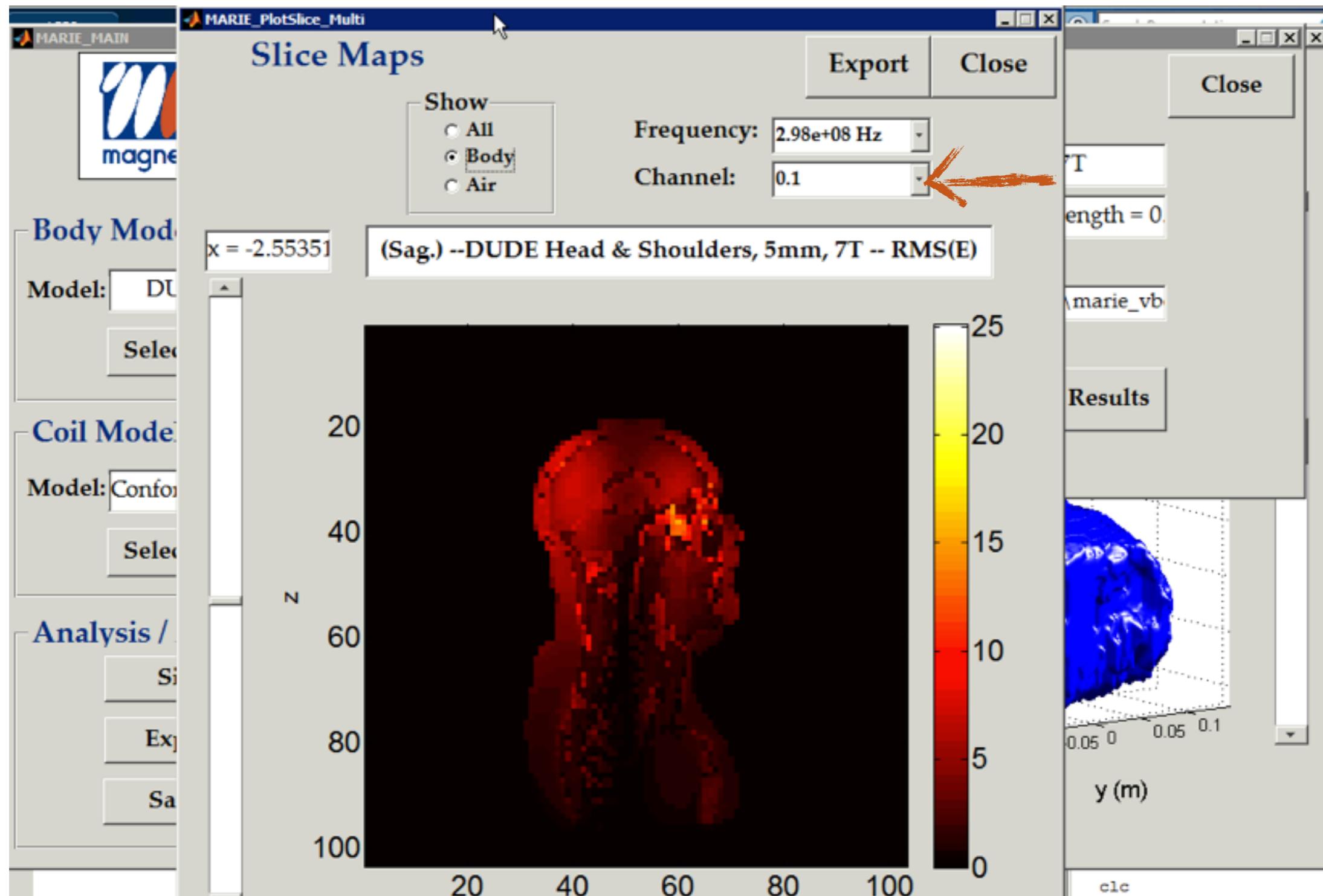
## ♦ Apply Pulse

- ♦ same interface as the main visualization window
  - ♦ select desired map and view, and plot



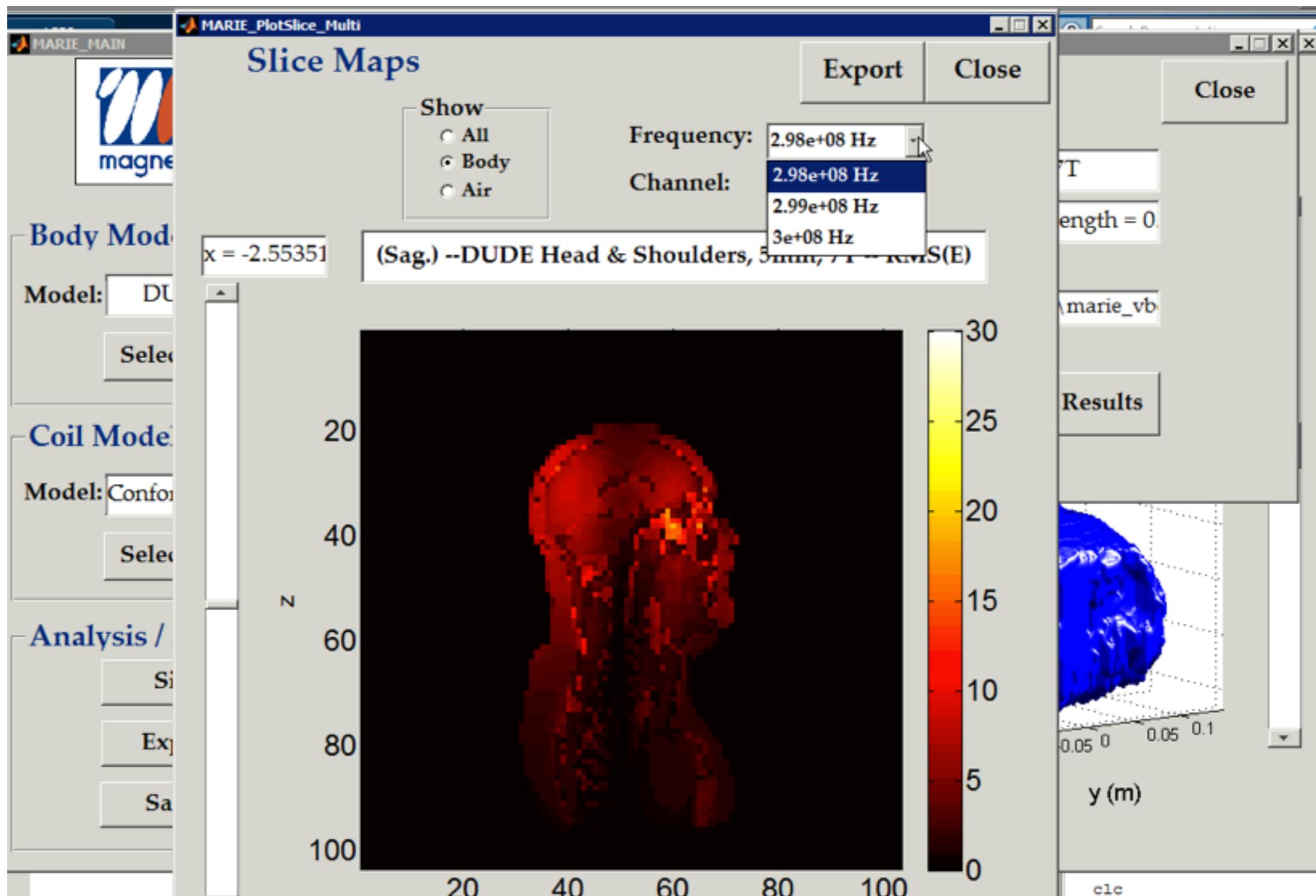
## ♦ Apply Pulse

- ♦ same plot interface, but instead of channels, we have the pulses



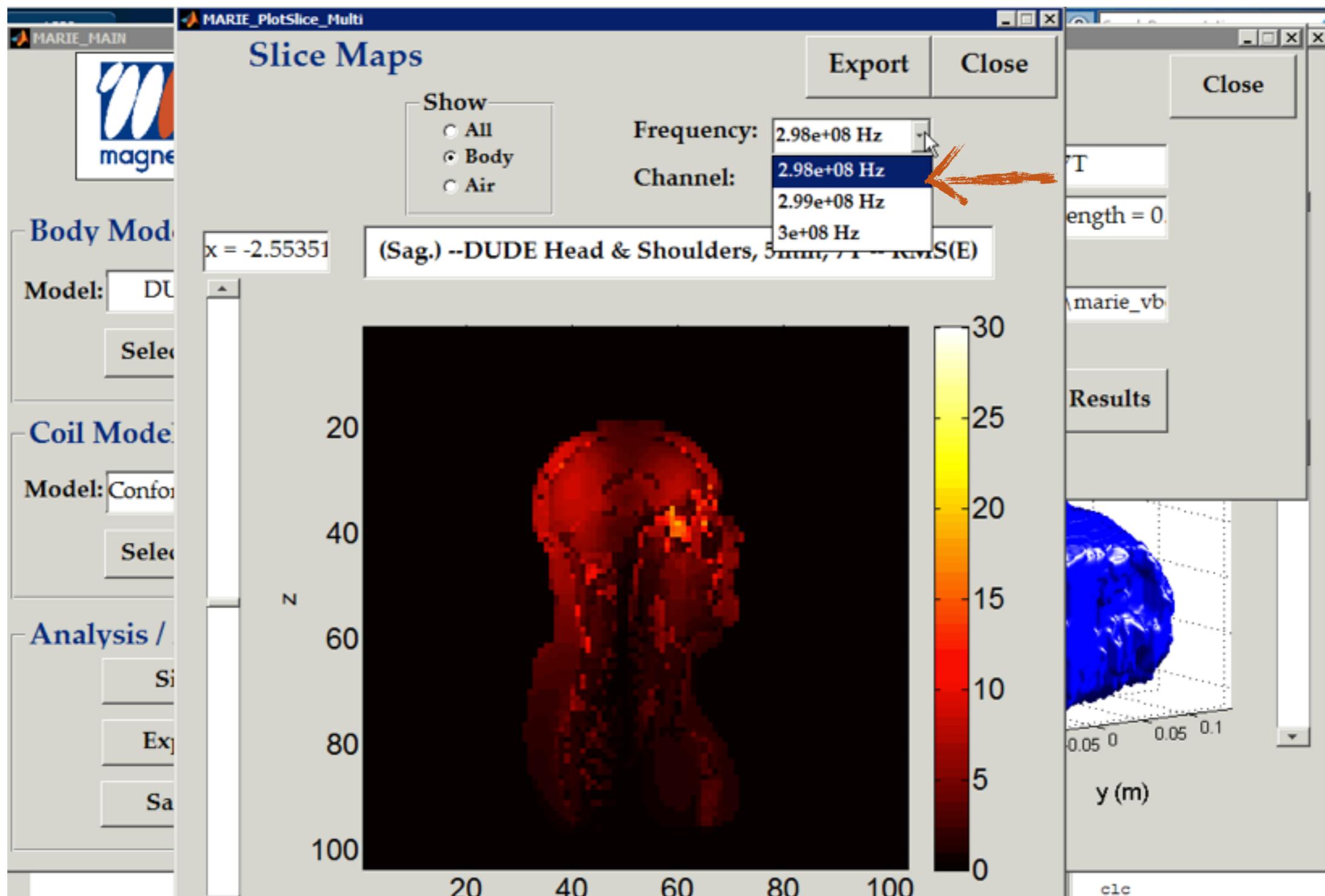
## ♦ Apply Pulse

- ♦ same plot interface, but instead of channels, we have the pulses



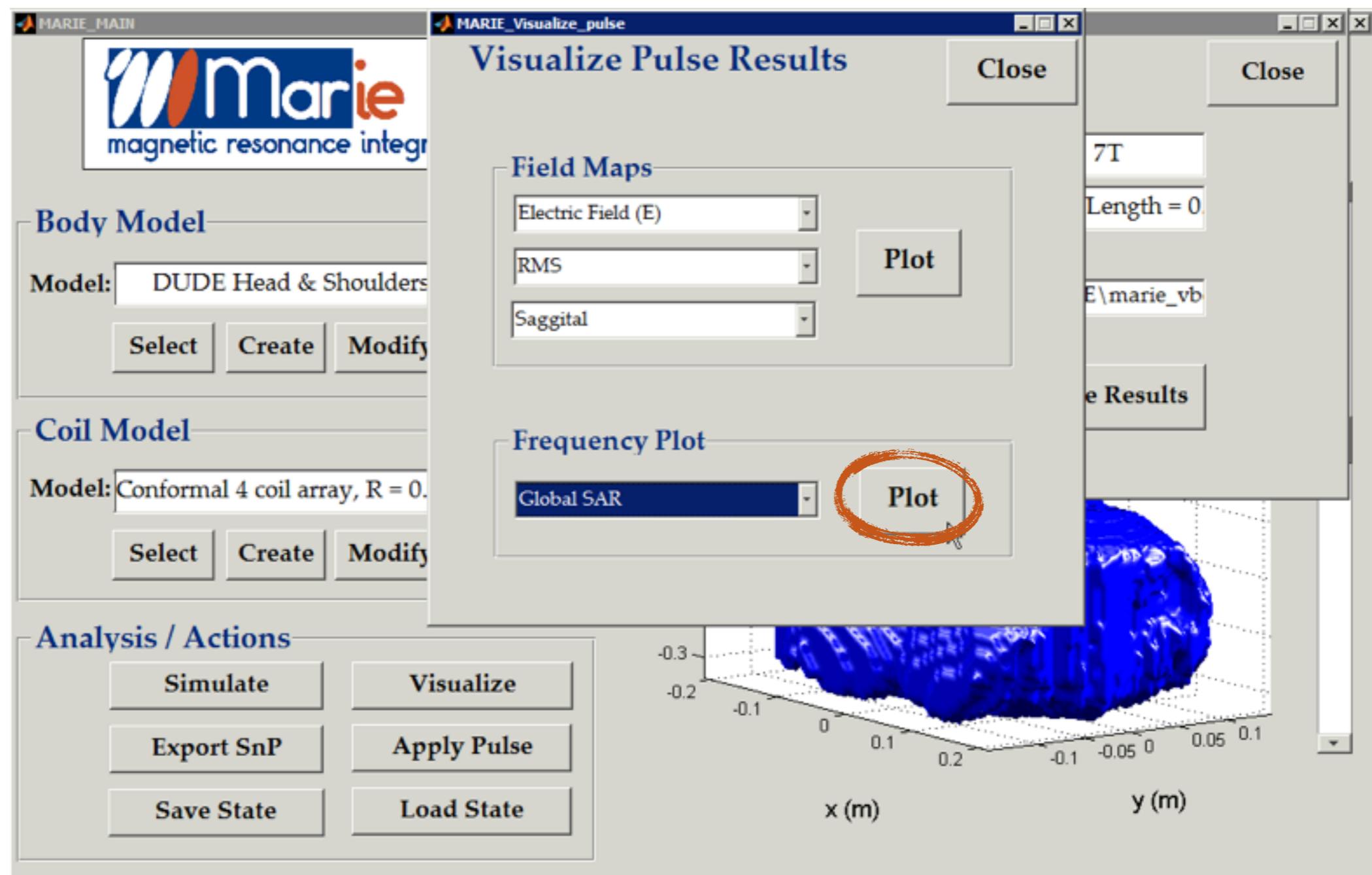
## ◆ Apply Pulse

- ◆ and the rest of features (frequency, cut position, etc...)



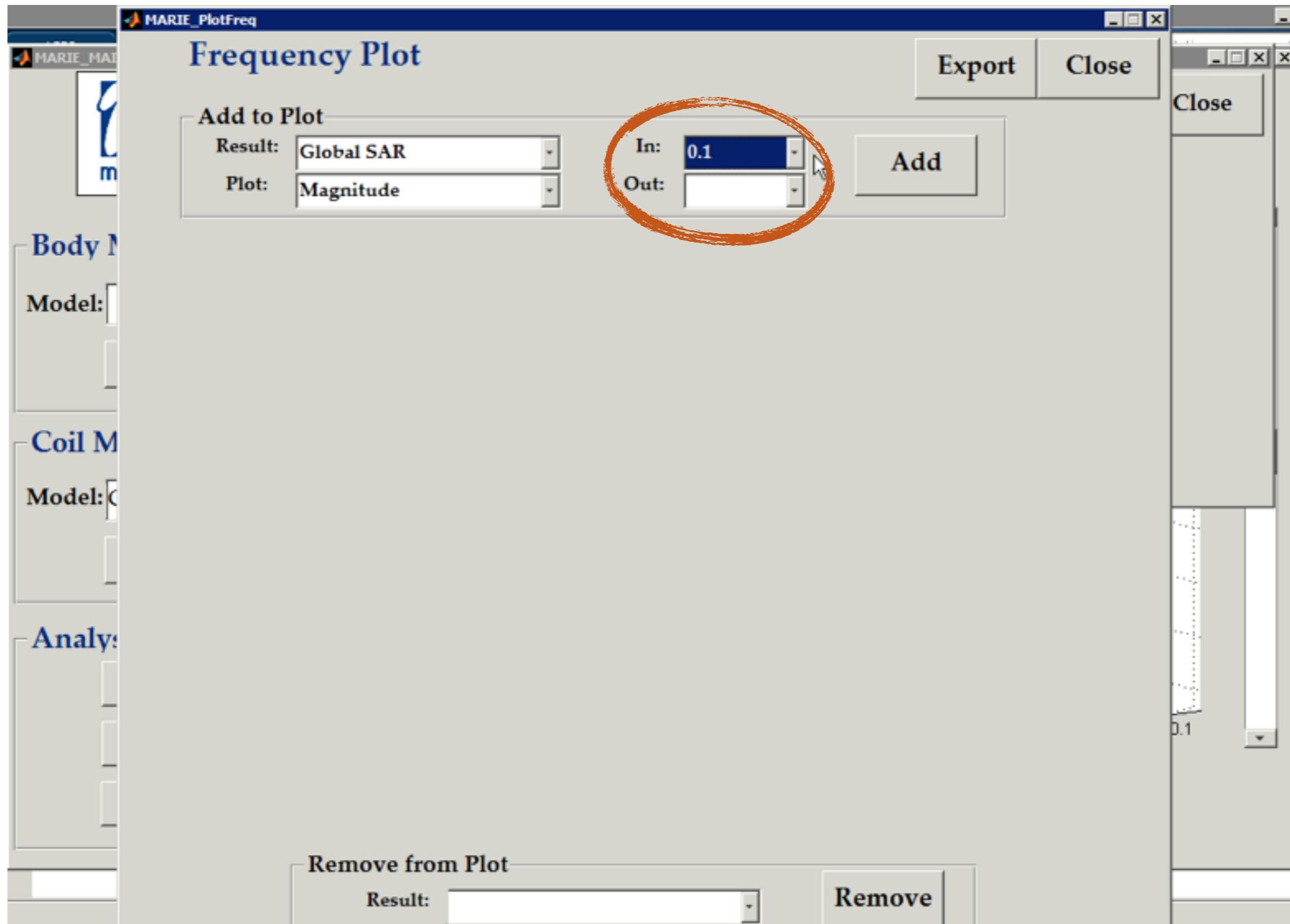
## ♦ Apply Pulse

- ♦ for the frequency dependent plots



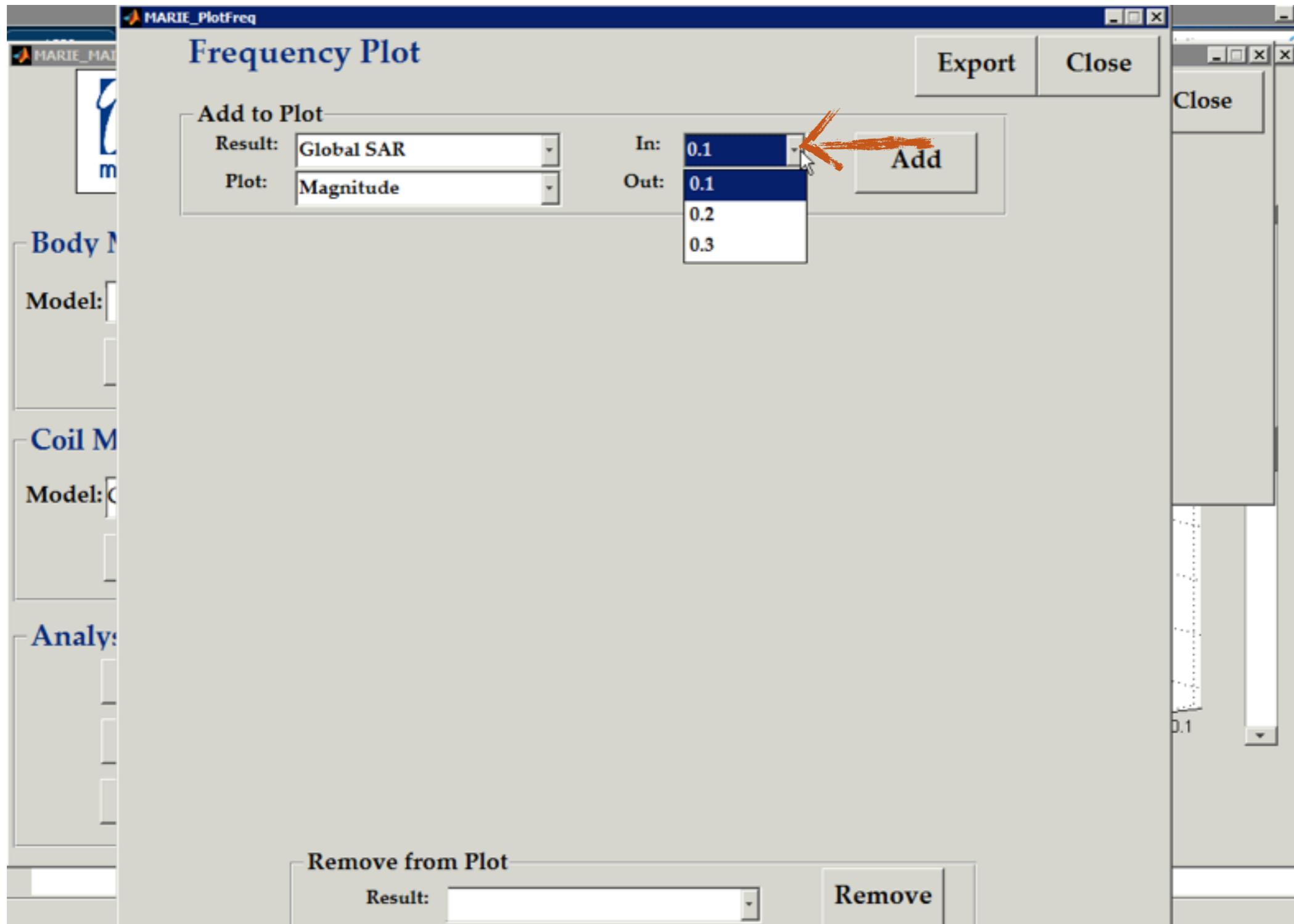
## ♦ Apply Pulse

- ♦ we have the same interface, with pulses instead of ports



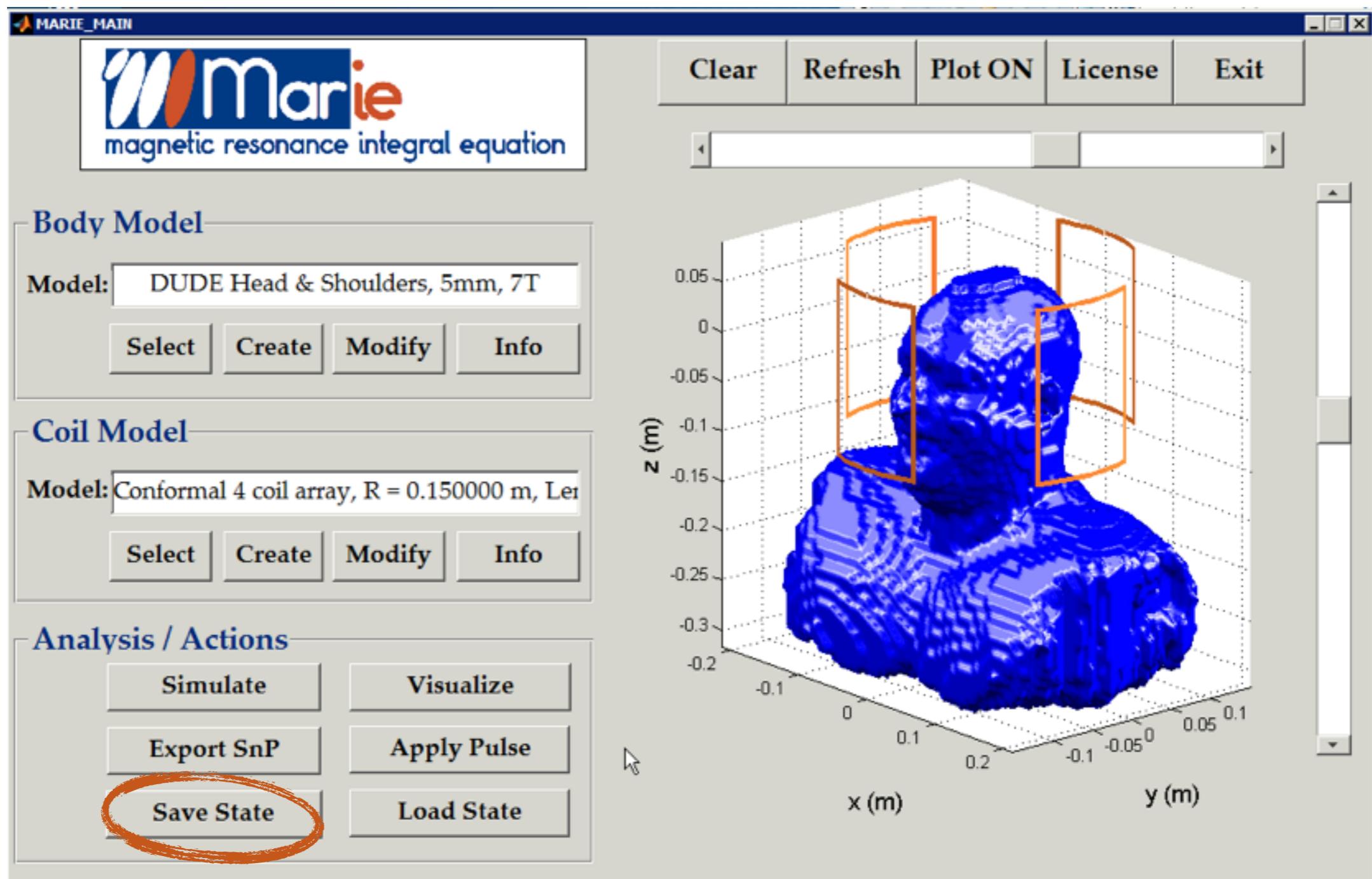
## ♦ Apply Pulse

- ♦ we have the same interface, with pulses instead of ports



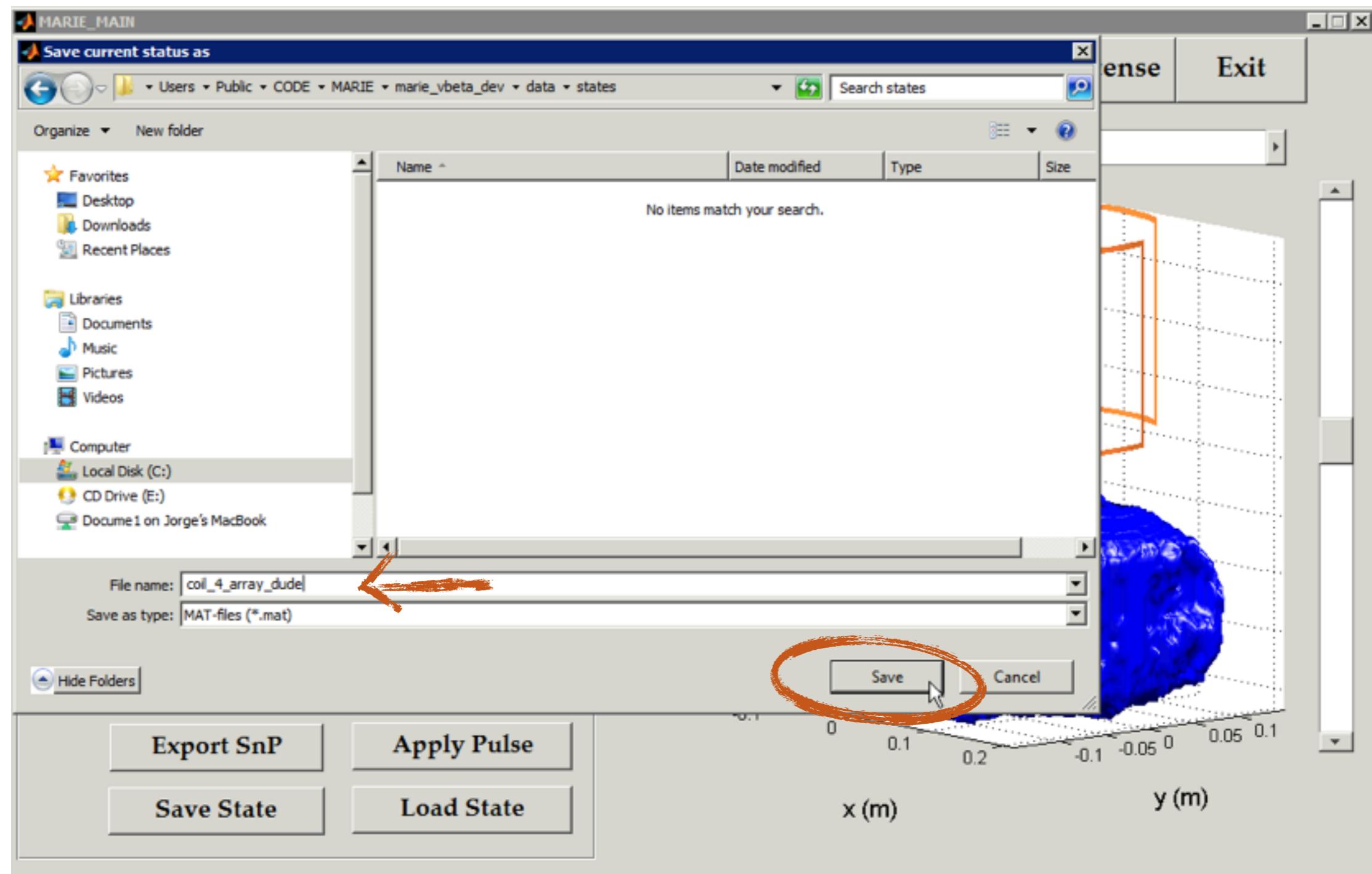
## ♦ Save the status

- ♦ “Save State” button saves the current workspace:
  - ♦ including RHBM, COIL, SOL and PULSE structures



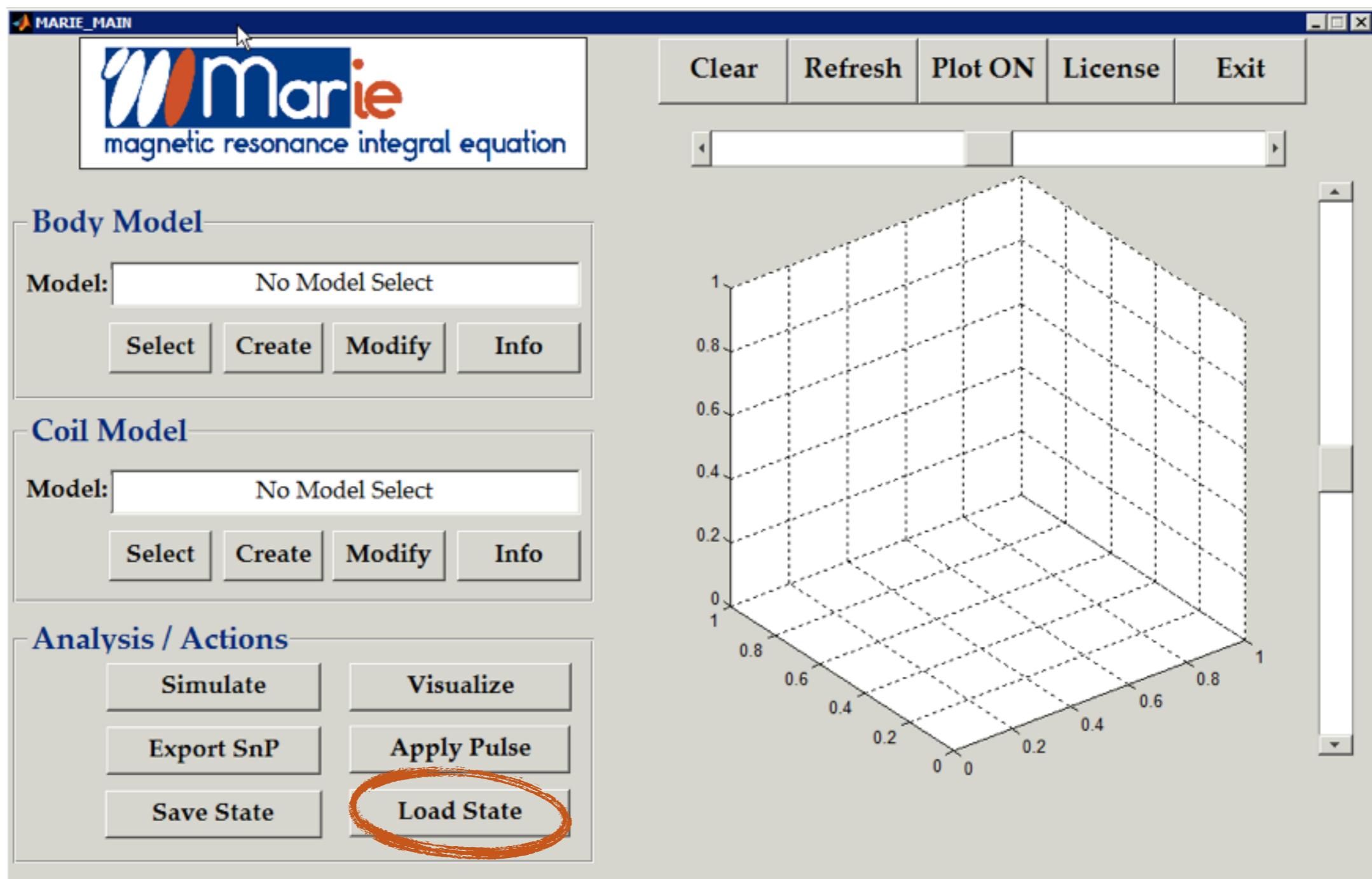
## ♦ Save the status

- ♦ a window dialogue will show: select folder and name
- ♦ press save, it will generate a .mat file, a window will confirm saving



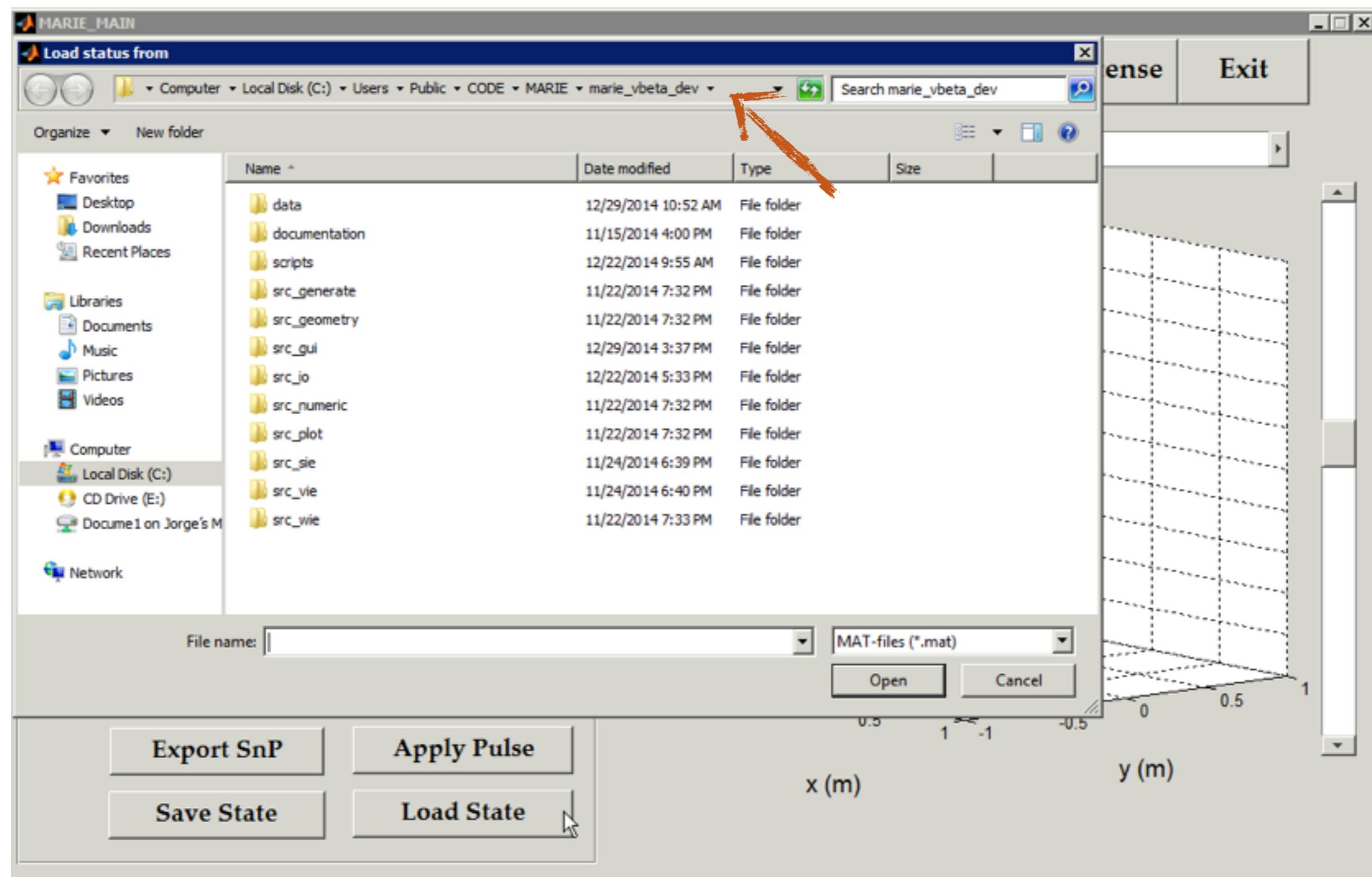
## ◆ Load a status

- ◆ “Load State” button loads a previously saved workspace:
  - ◆ including RHBM, COIL, SOL and PULSE structures



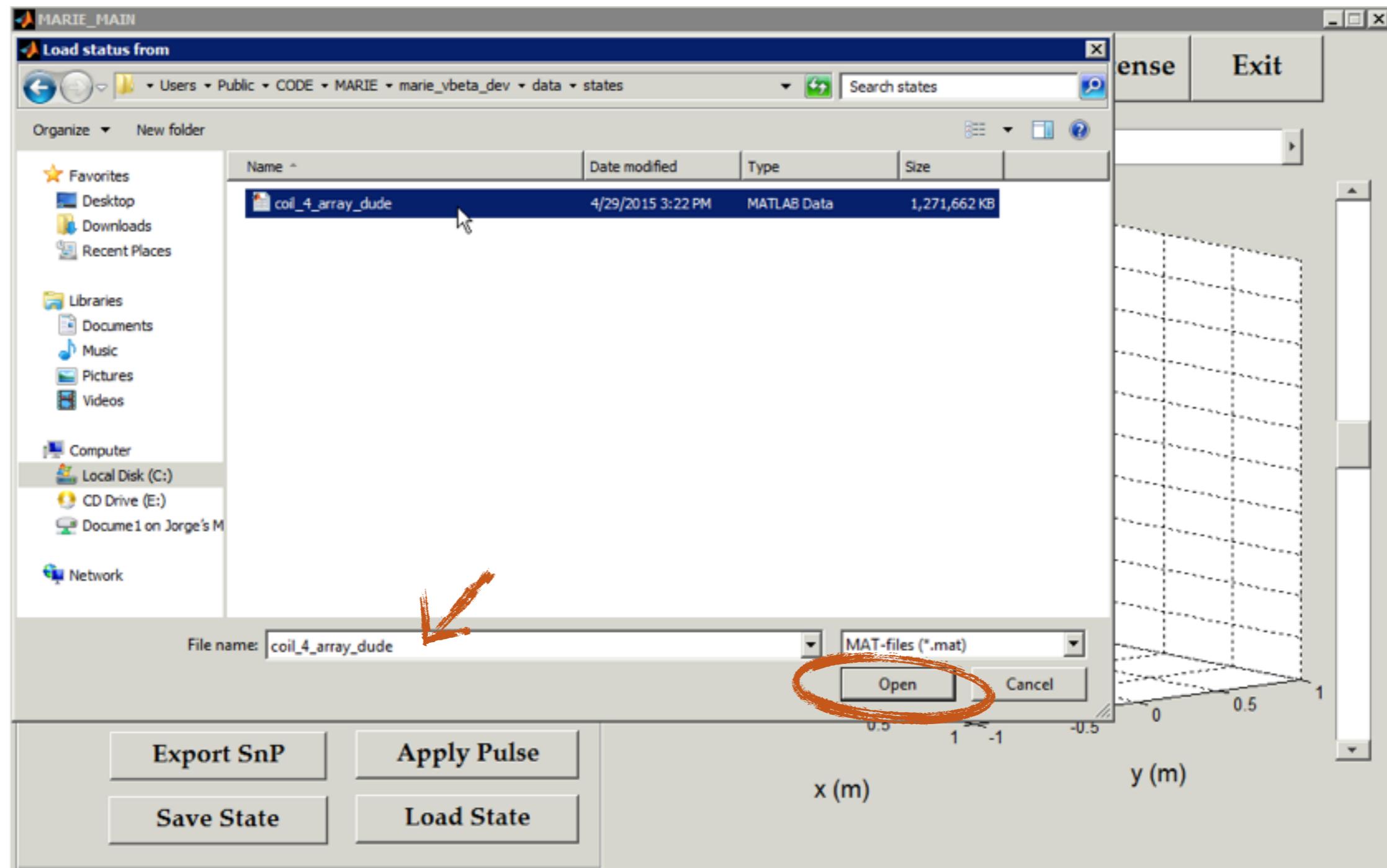
## ◆ Load a status

- ◆ a window dialogue will show: navigate to desired file

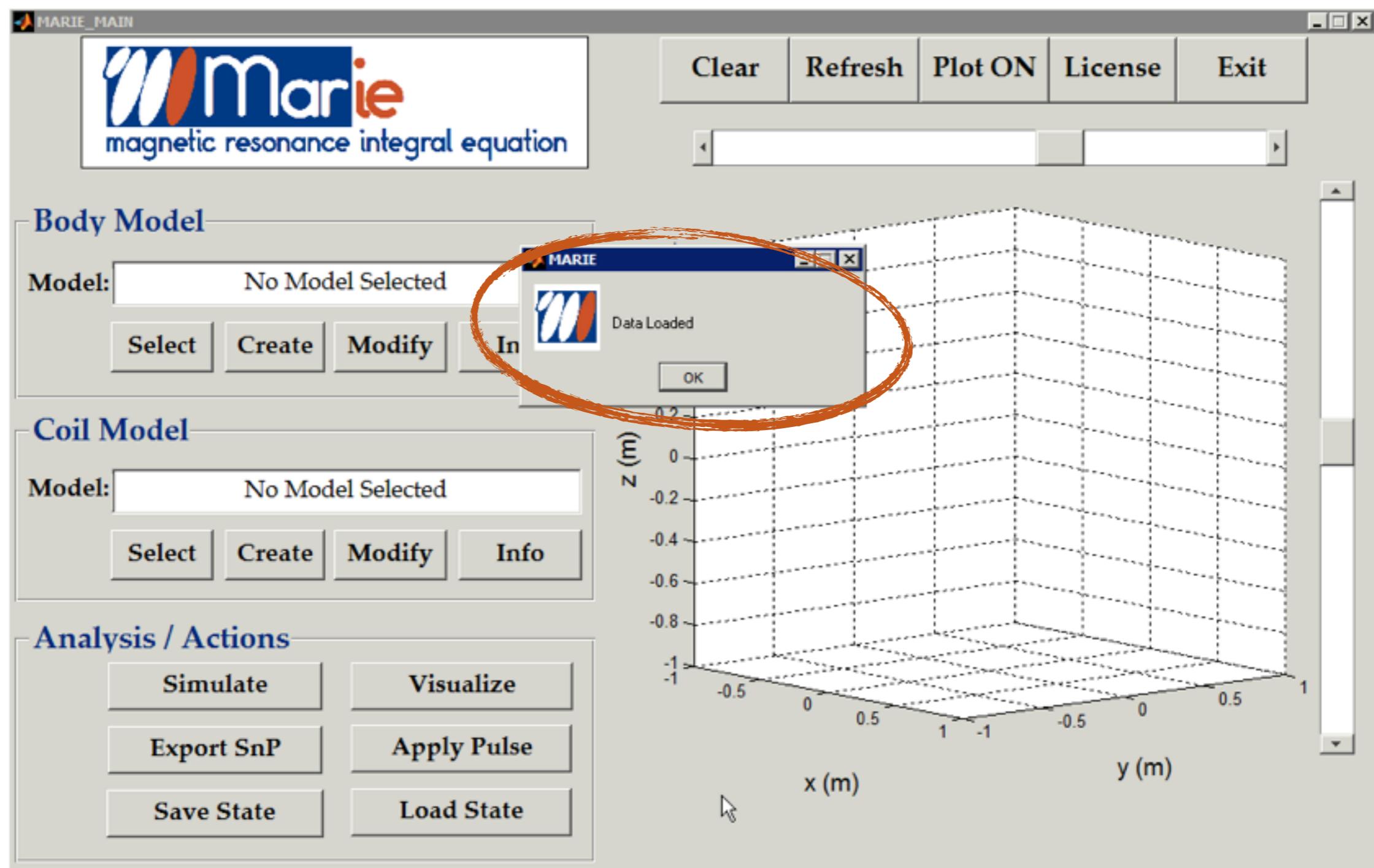


## ◆ Load a status

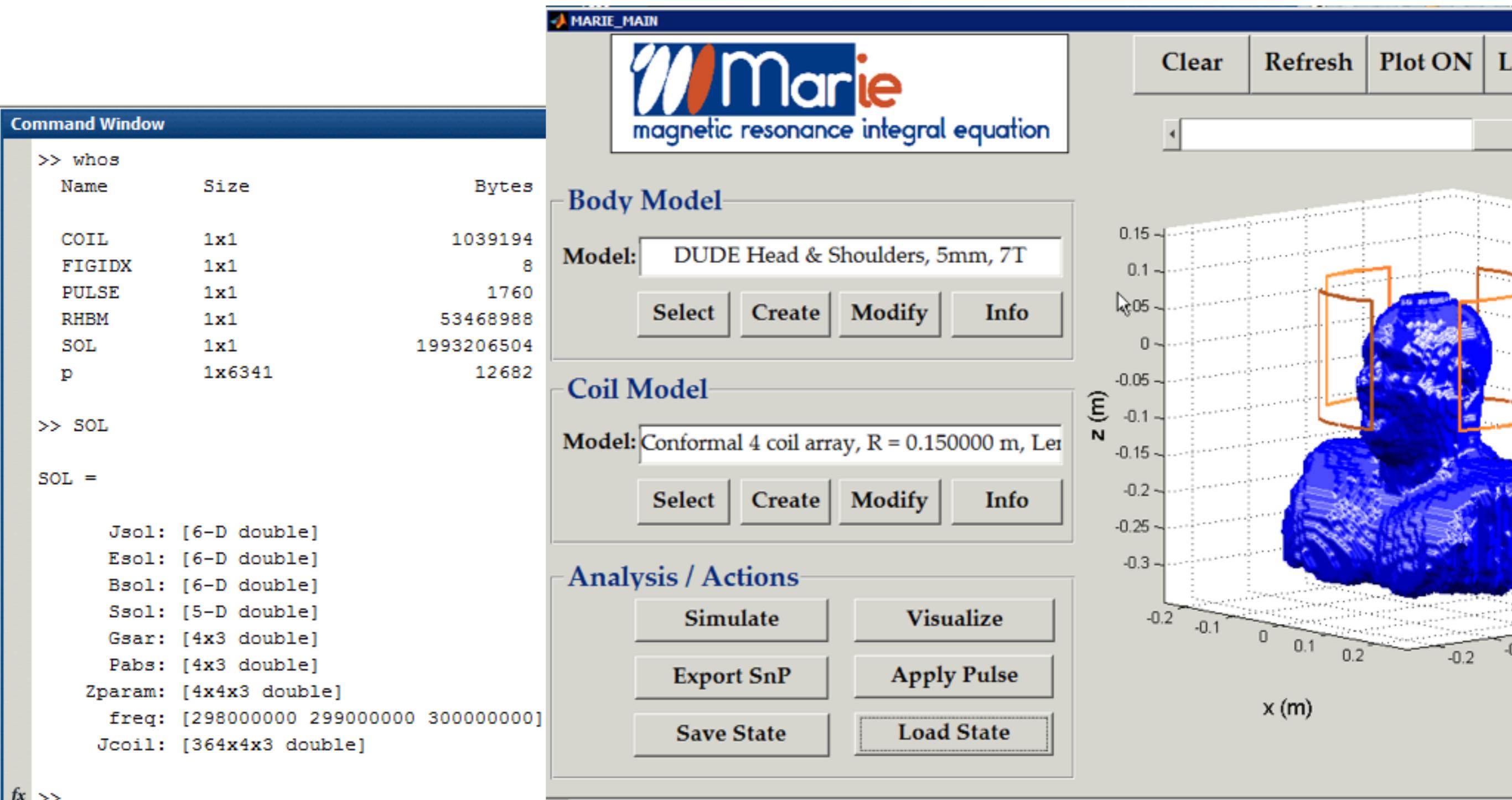
- ◆ a window dialogue will show: navigate to desired file
- ◆ select the .mat file, and press open



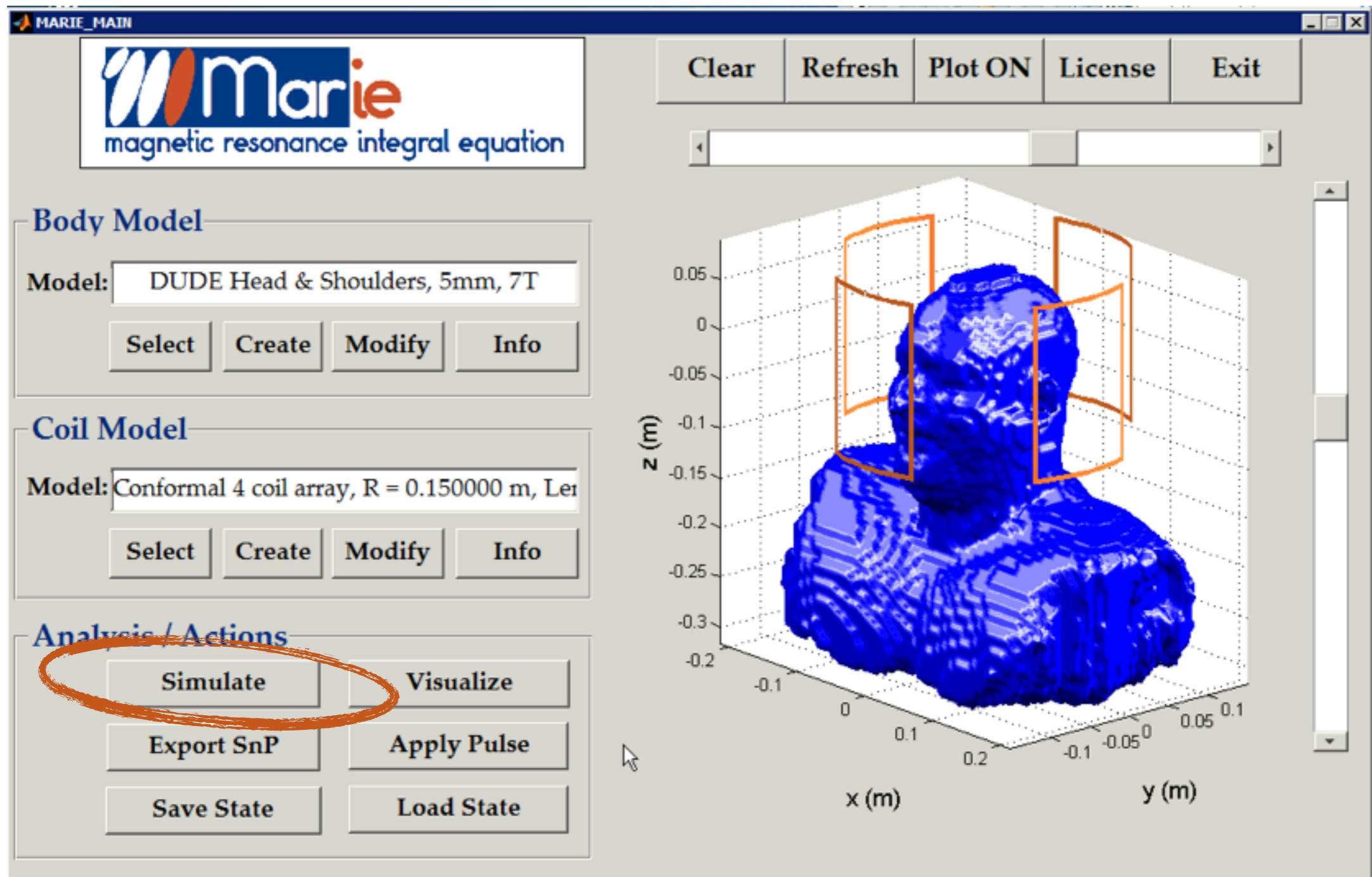
- ◆ Load a status
  - ◆ a window will confirm the loading



- ◆ Load a status
  - ◆ data will be automatically loaded

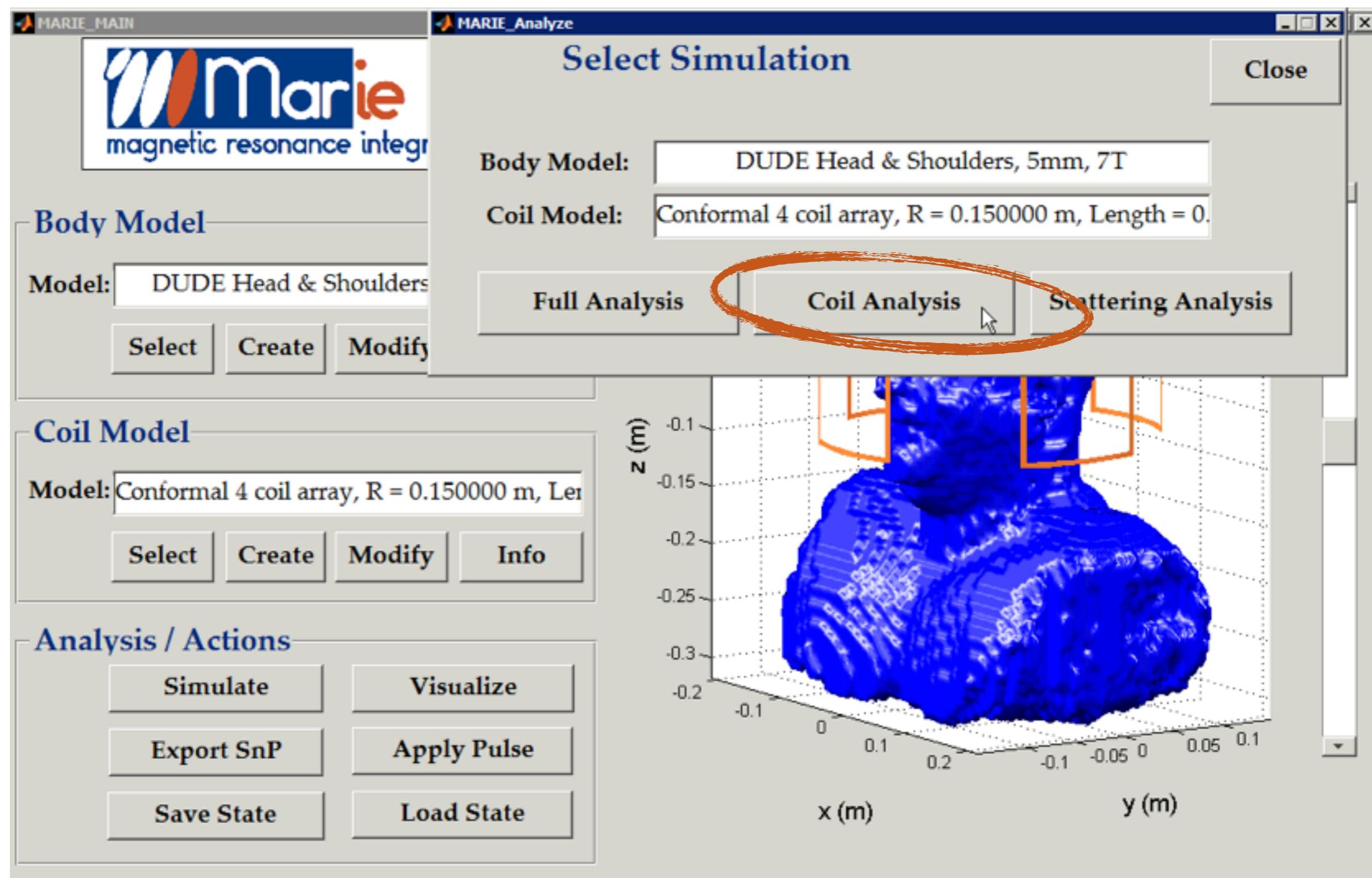


- ♦ Call the simulation engine
  - ♦ push “simulate” button



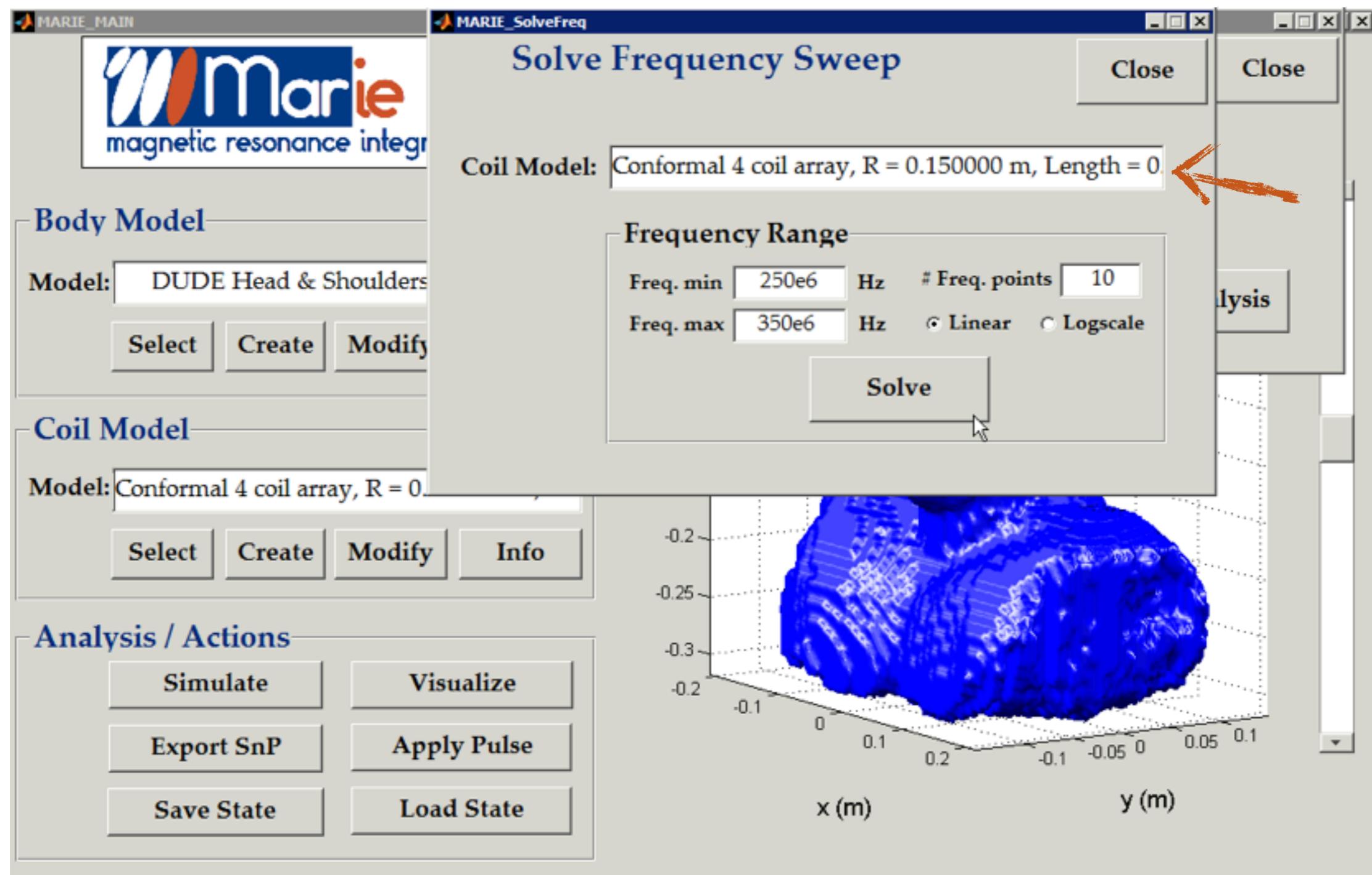
## ◆ Coil analysis

- ◆ push the “coil analysis” button



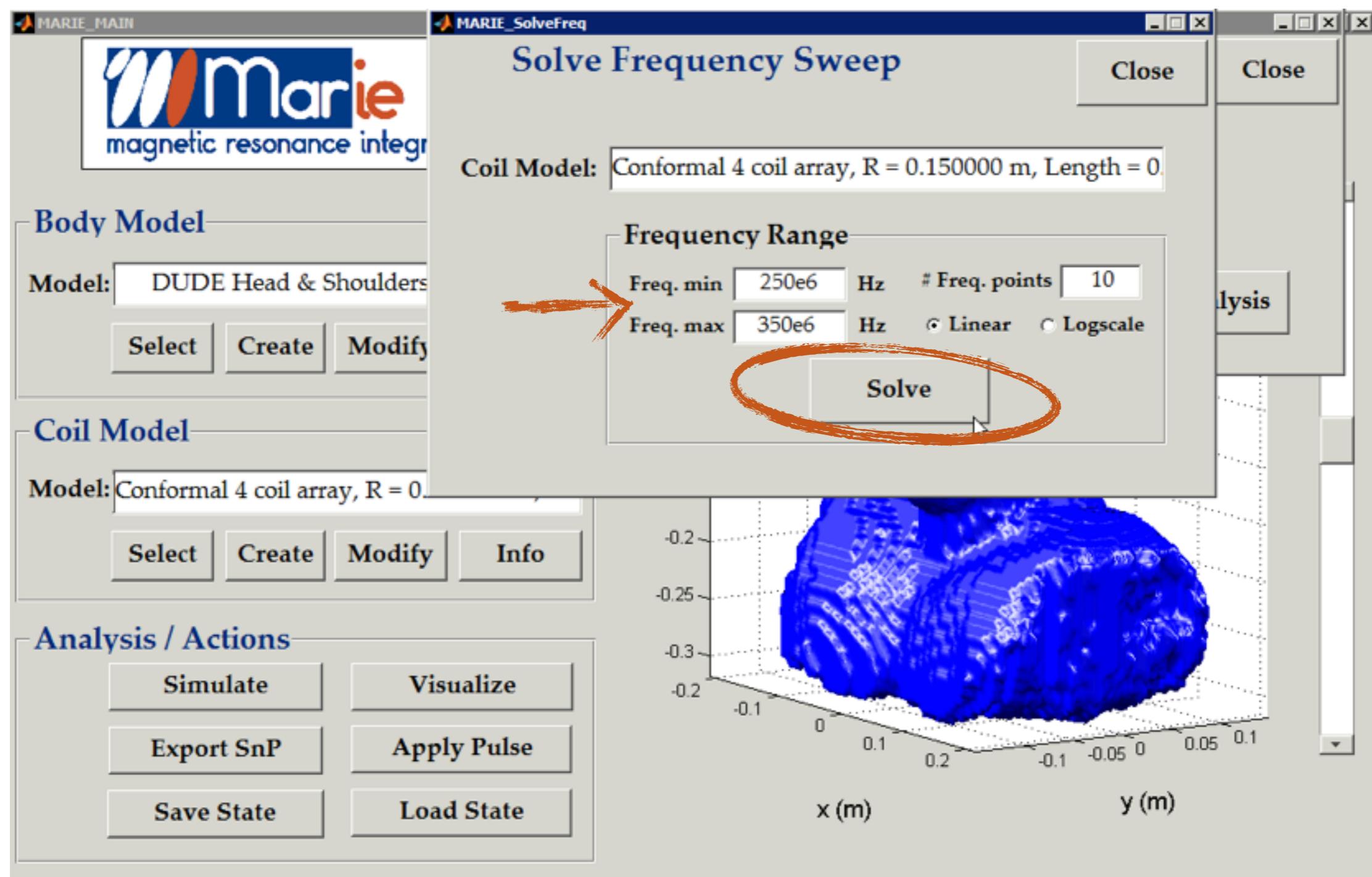
## ♦ Coil analysis

- ♦ opens the coil frequency sweep dialogue
- ♦ coil model name is presented



## ♦ Coil analysis

- ♦ select the frequency range, number of points and distribution
- ♦ push the “solve” button to run the simulation



## ♦ Coil analysis

- ♦ simulation info will appear in command line

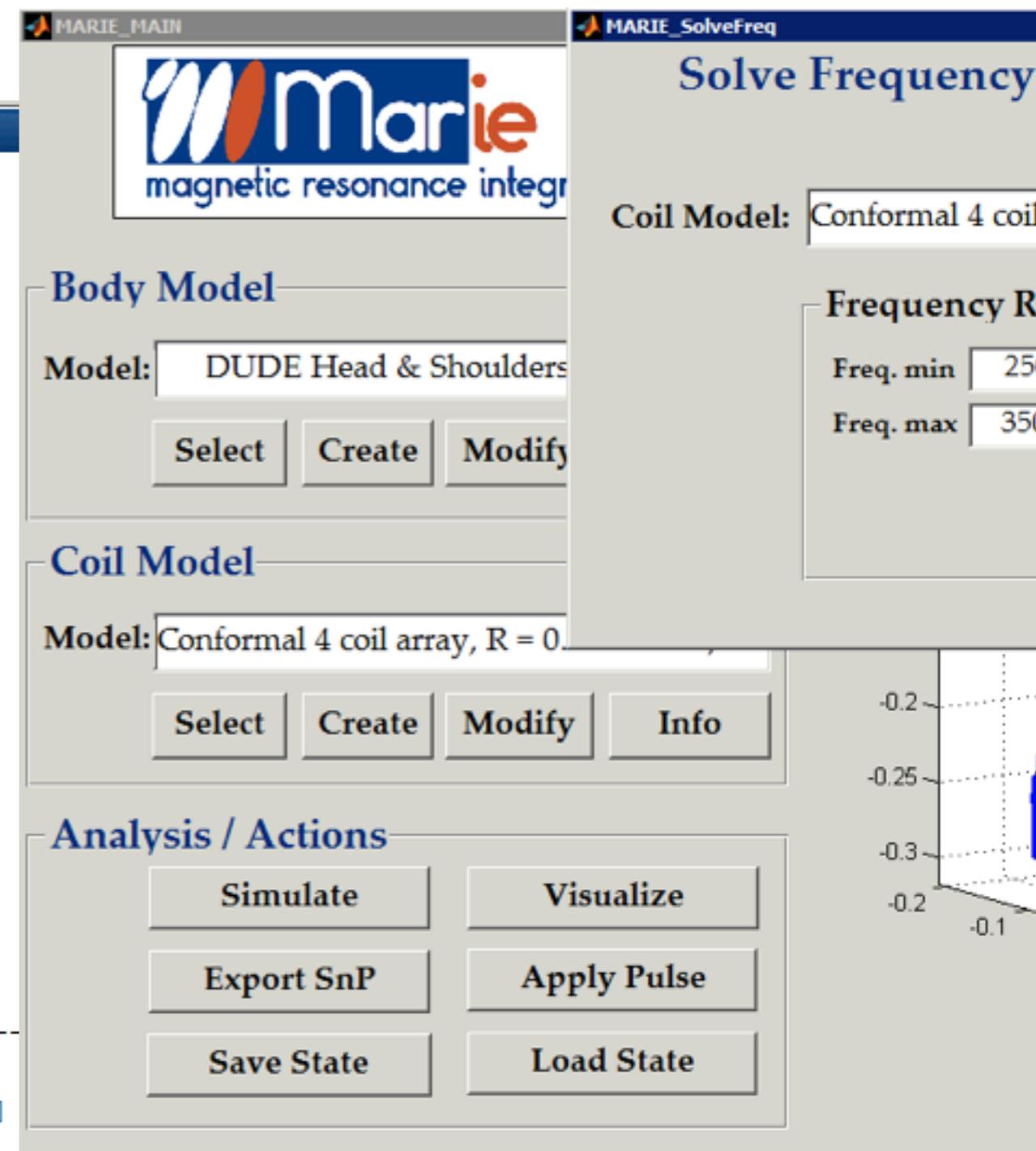
**Command Window**

```
# DOFS: 364
# PORTS: 4
# NS interactions: 58175
# EA interactions: 356
# VA interactions: 465
# ST interactions: 344
Operating Frequency: 350.000 MHz

Time in NS: 2.80 sec
Time in EA: 0.36 sec
Time in VA: 0.20 sec
Time in ST: 0.27 sec
Time in Port: 0.00 sec

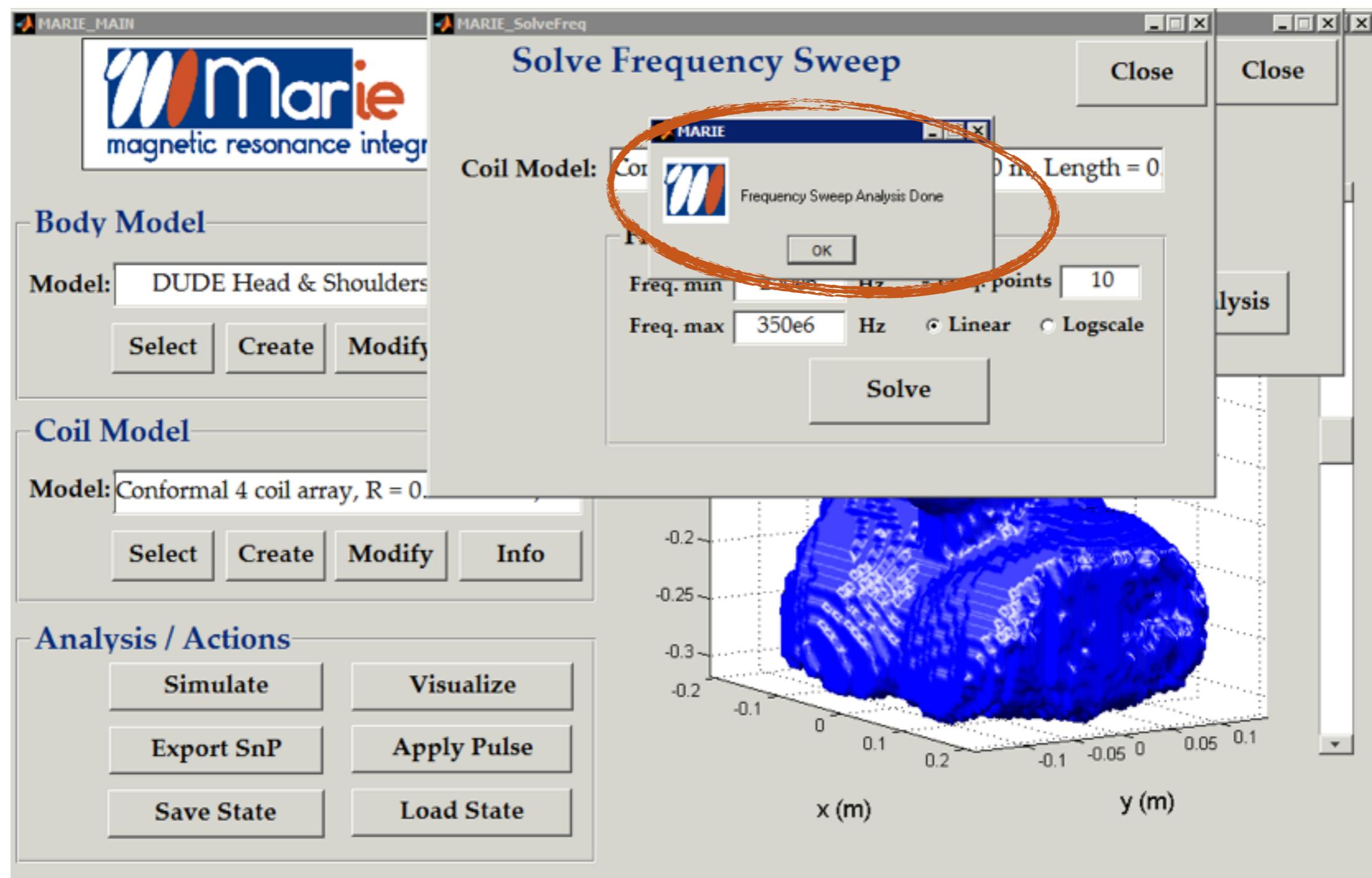
Overall TIME: 3.65 sec

-----
SIE system solve done, Elapsed time = 0.01 [sec]
-----
SIE frequency sweep for 10 freqs. done, overall time 37.76 [sec]
```



## ♦ Coil analysis

- ♦ simulation info will appear in command line
- ♦ window will appear when analysis is done



## ♦ Coil analysis

- ♦ the results will be stored in SOL structure
- ♦ only coil related results are generated and stored
  - ♦ remember: free space simulation!

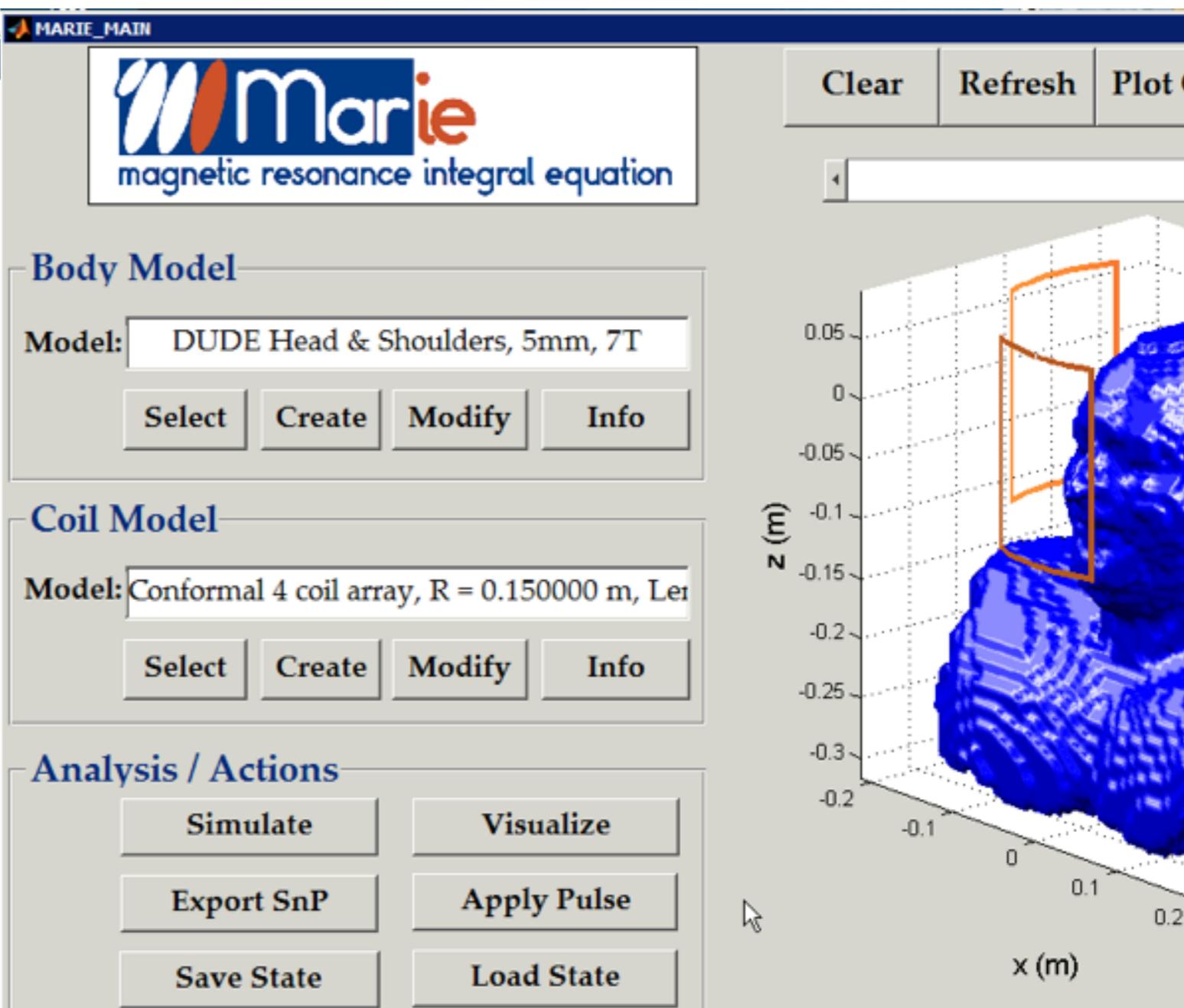
Command Window

```
>> whos
  Name      Size            Bytes  Class
  COIL      1x1          1039194  struct
  FIGIDX    1x1                  8  double
  PULSE     1x1          1760   struct
  RHBM      1x1        53468988  struct
  SOL       1x1          237184  struct
  p          1x6341        12682  char

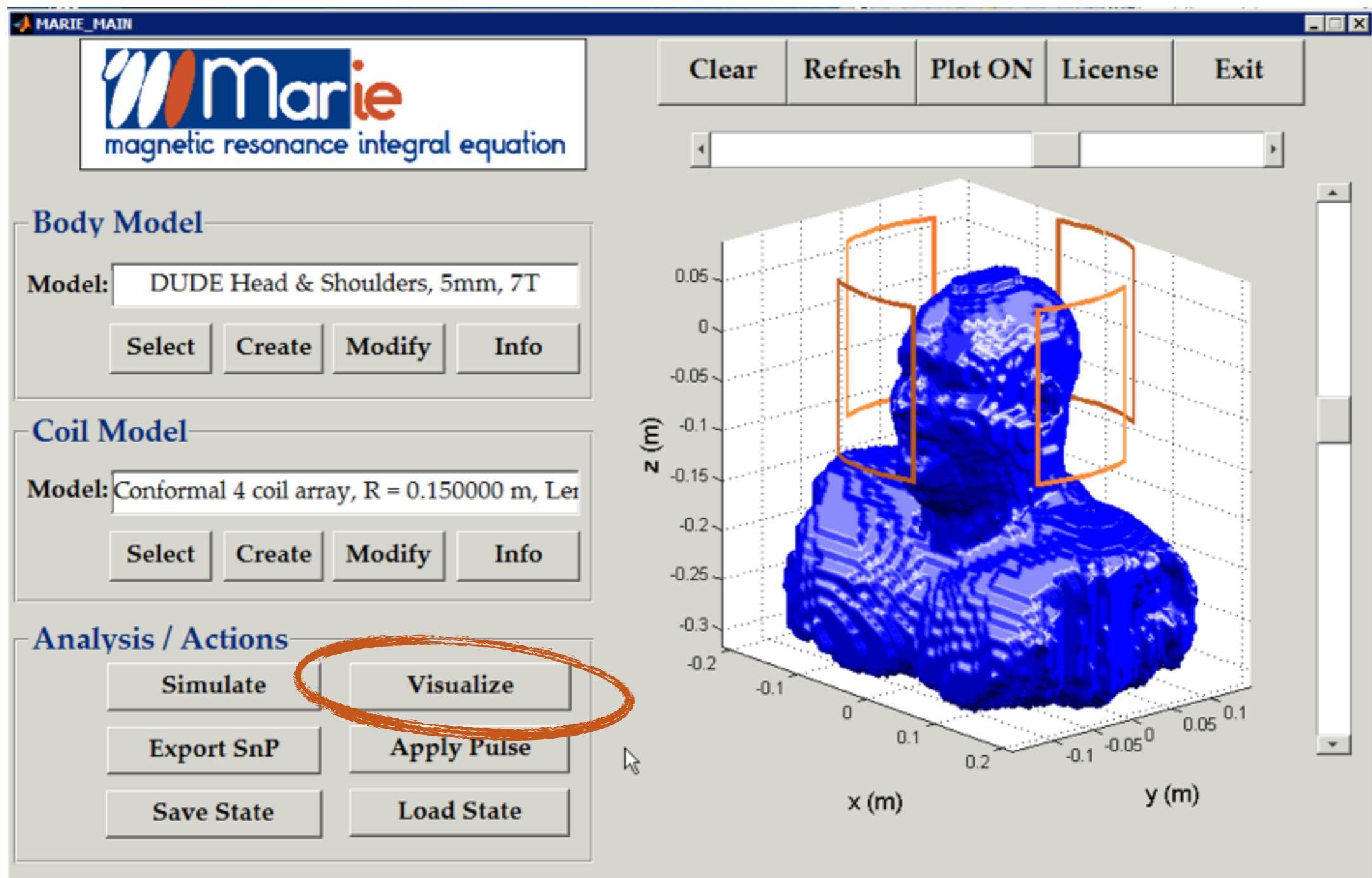
>> SOL

SOL =
  Jsol: []
  Esol: []
  Bsol: []
  Ssol: []
  Gsar: []
  Pabs: []
  Zparam: [4x4x10 double]
  freq: [1x10 double]
  Jcoil: [364x4x10 double]
```

fx >>

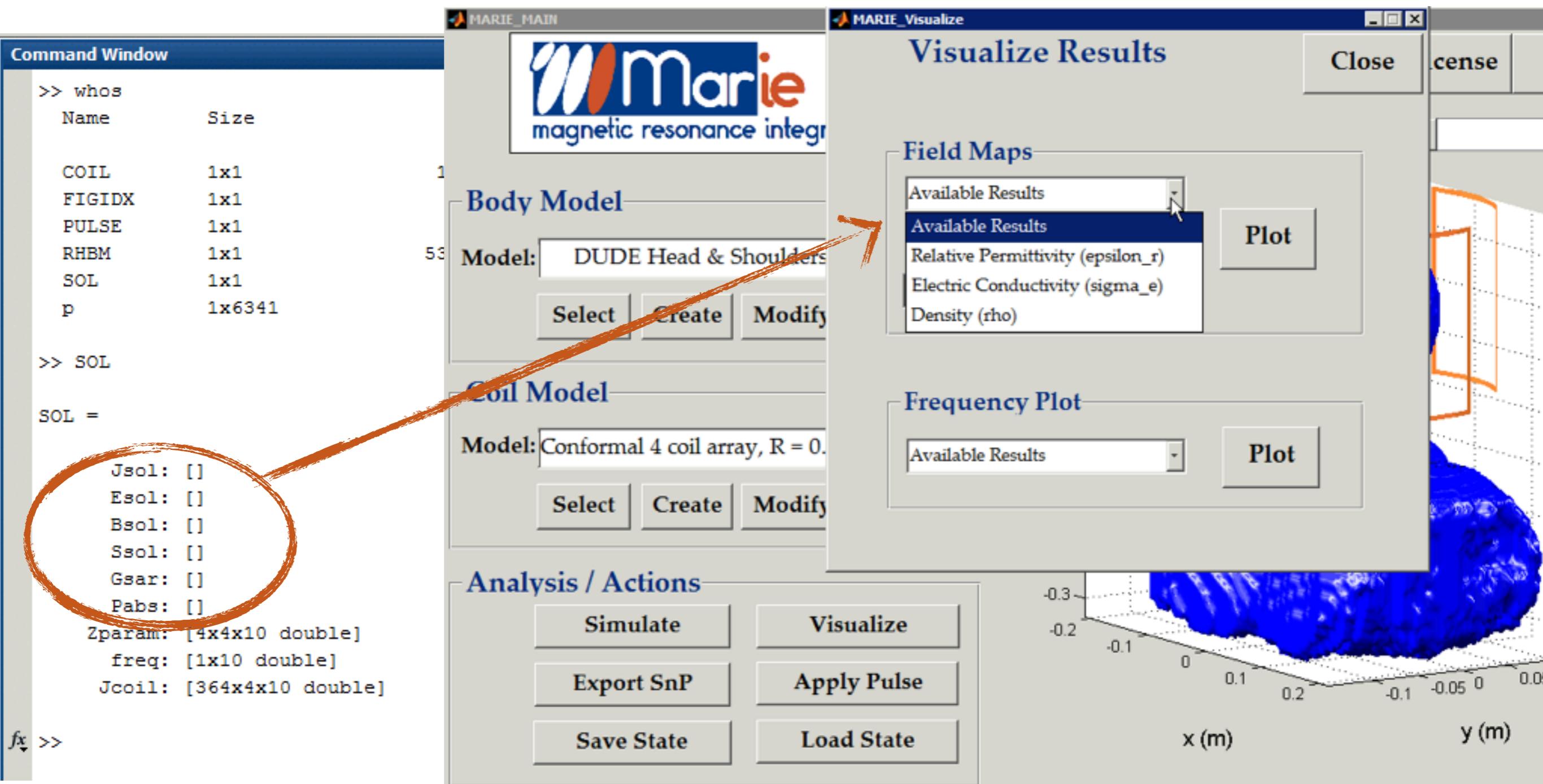


- ◆ Visualize the results
  - ◆ push the “visualize” button



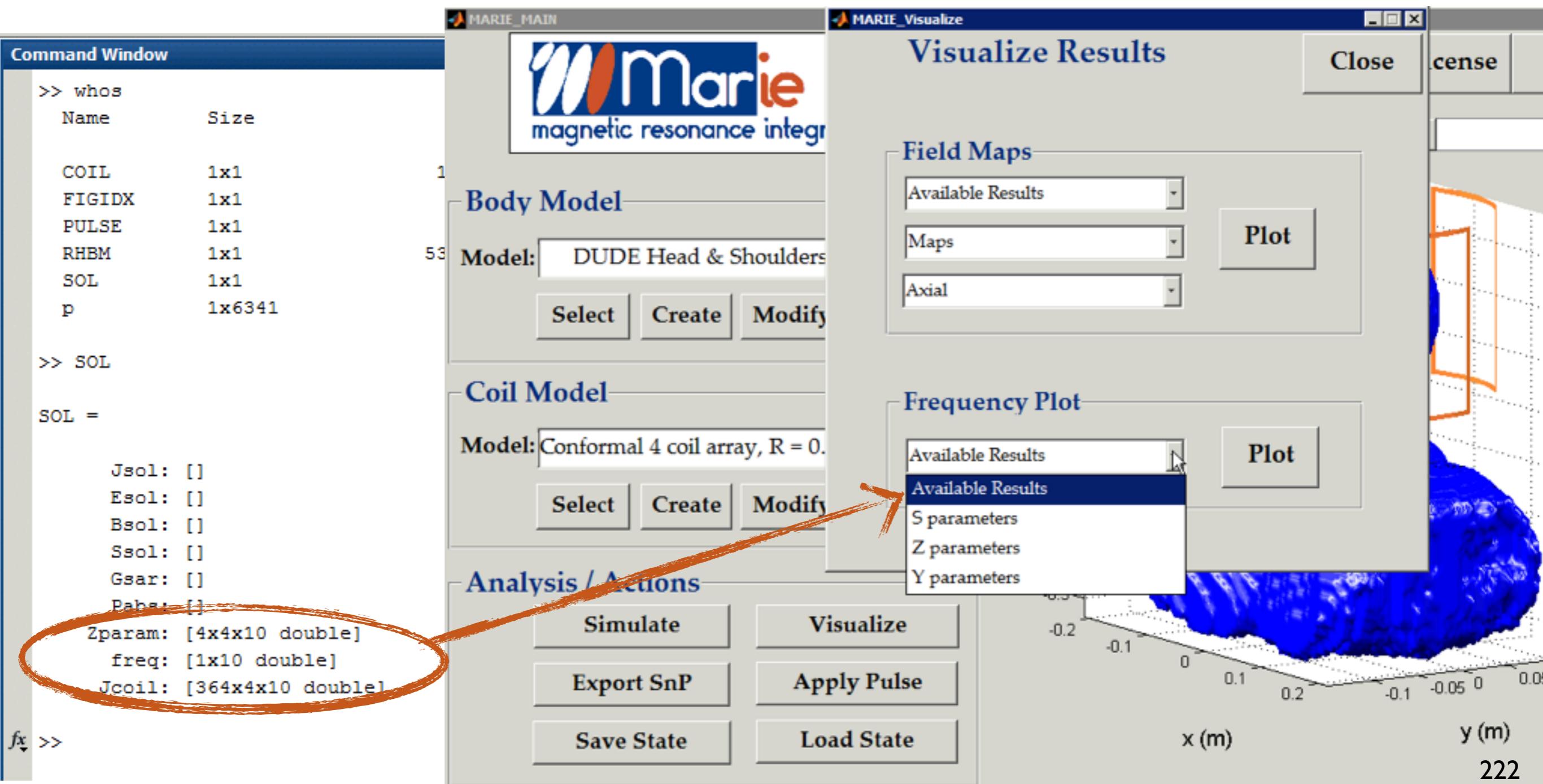
## ♦ Visualize the results

- ♦ only coil related results are available
  - ♦ no fields or body currents, no SAR and no absorbed power

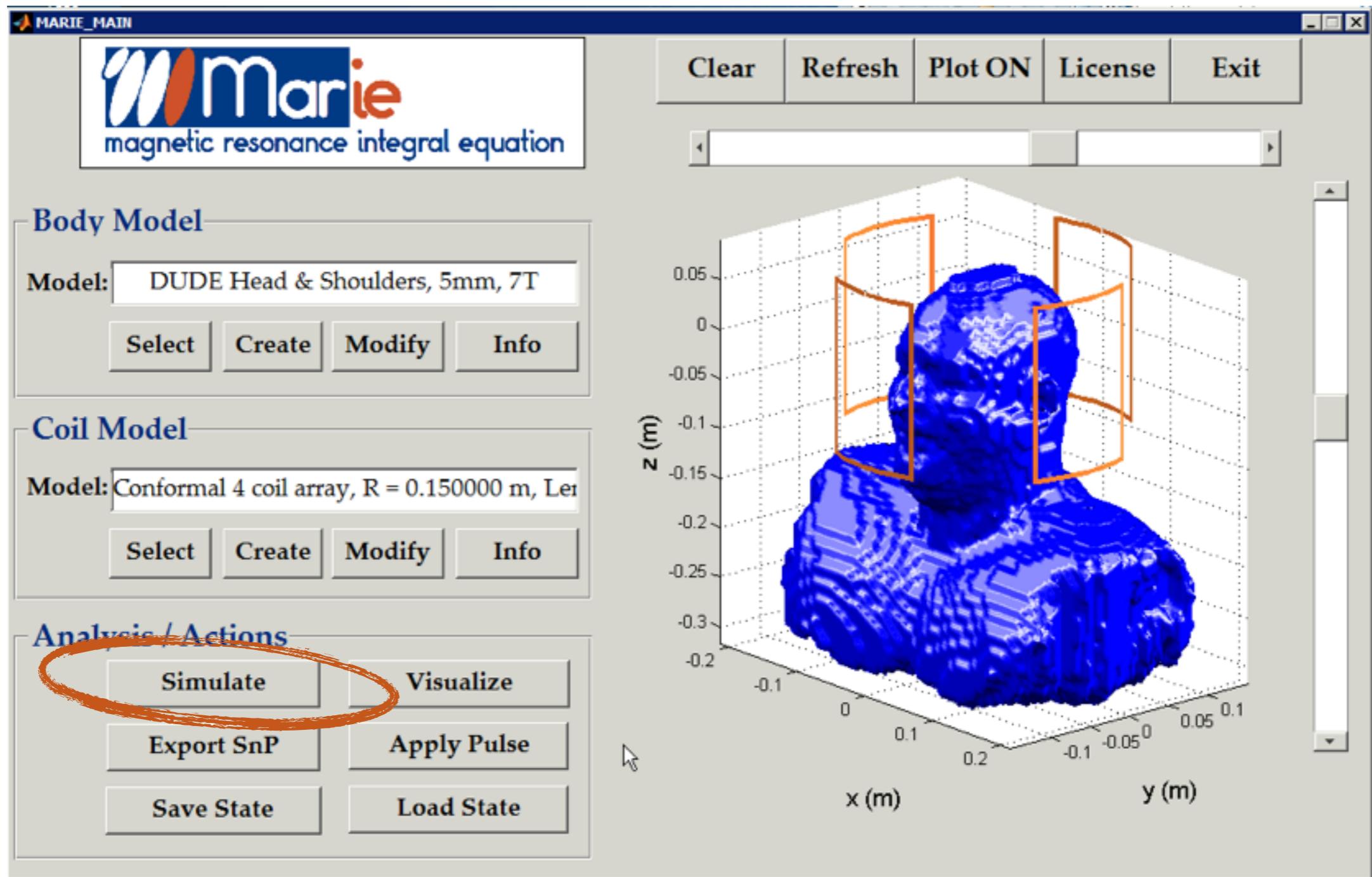


## ♦ Visualize the results

- ♦ only coil related results are available
  - ♦ free-space coil port parameters are available

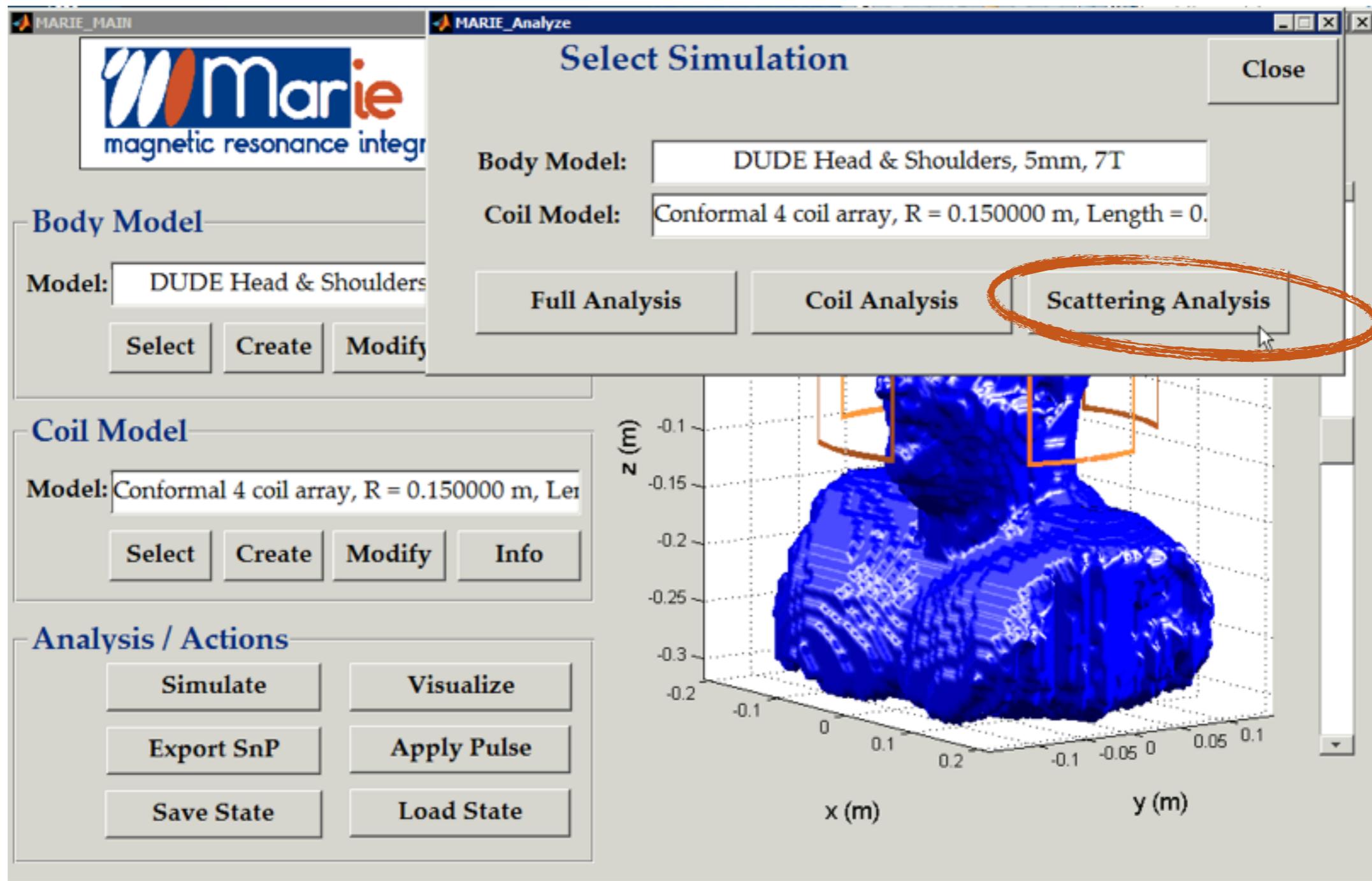


- ♦ Call the simulation engine
  - ♦ push “simulate” button



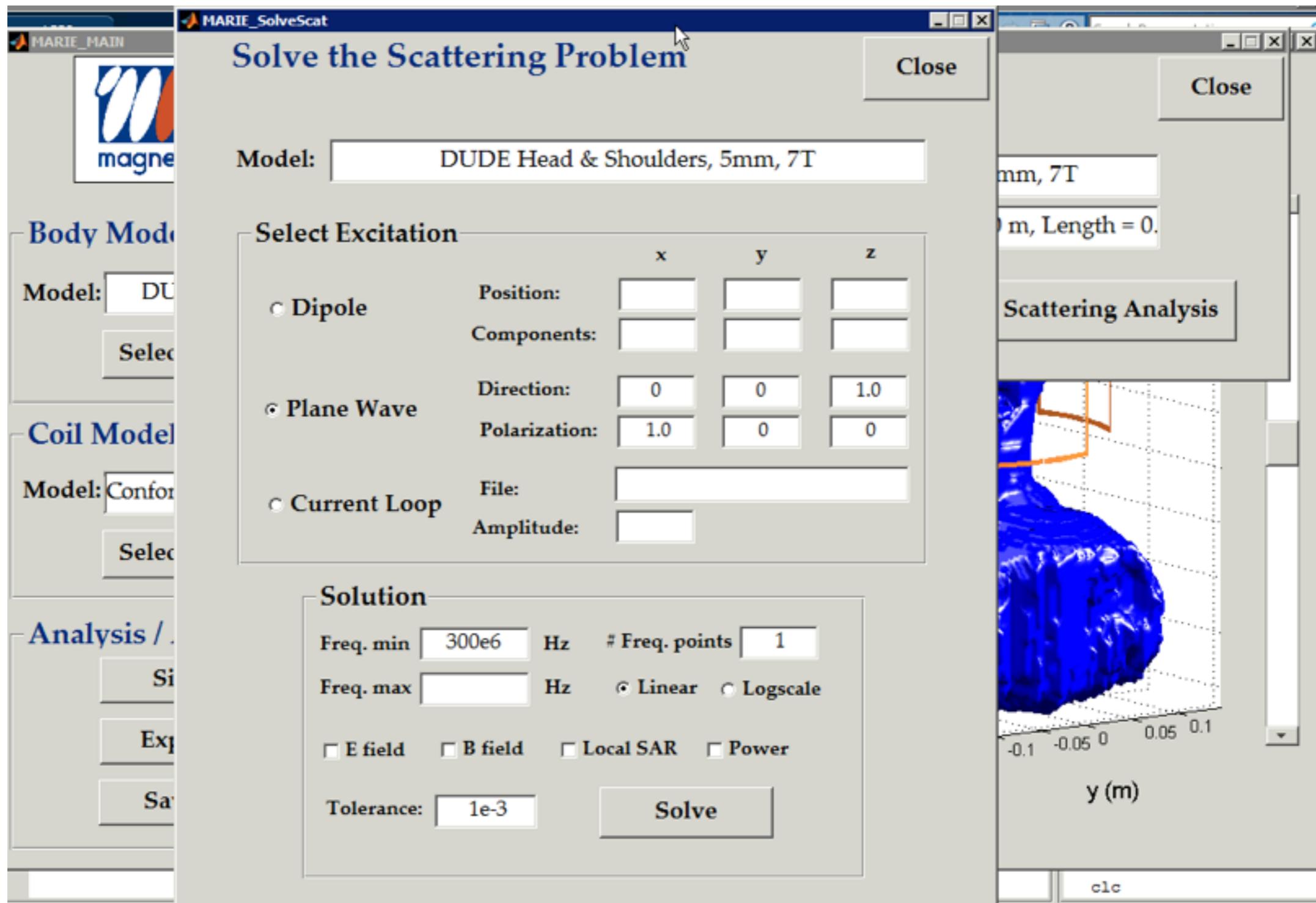
## ♦ Scattering analysis

- ♦ solves the scattering problem with the body due to some excitation
- ♦ this analysis ignores the coil model

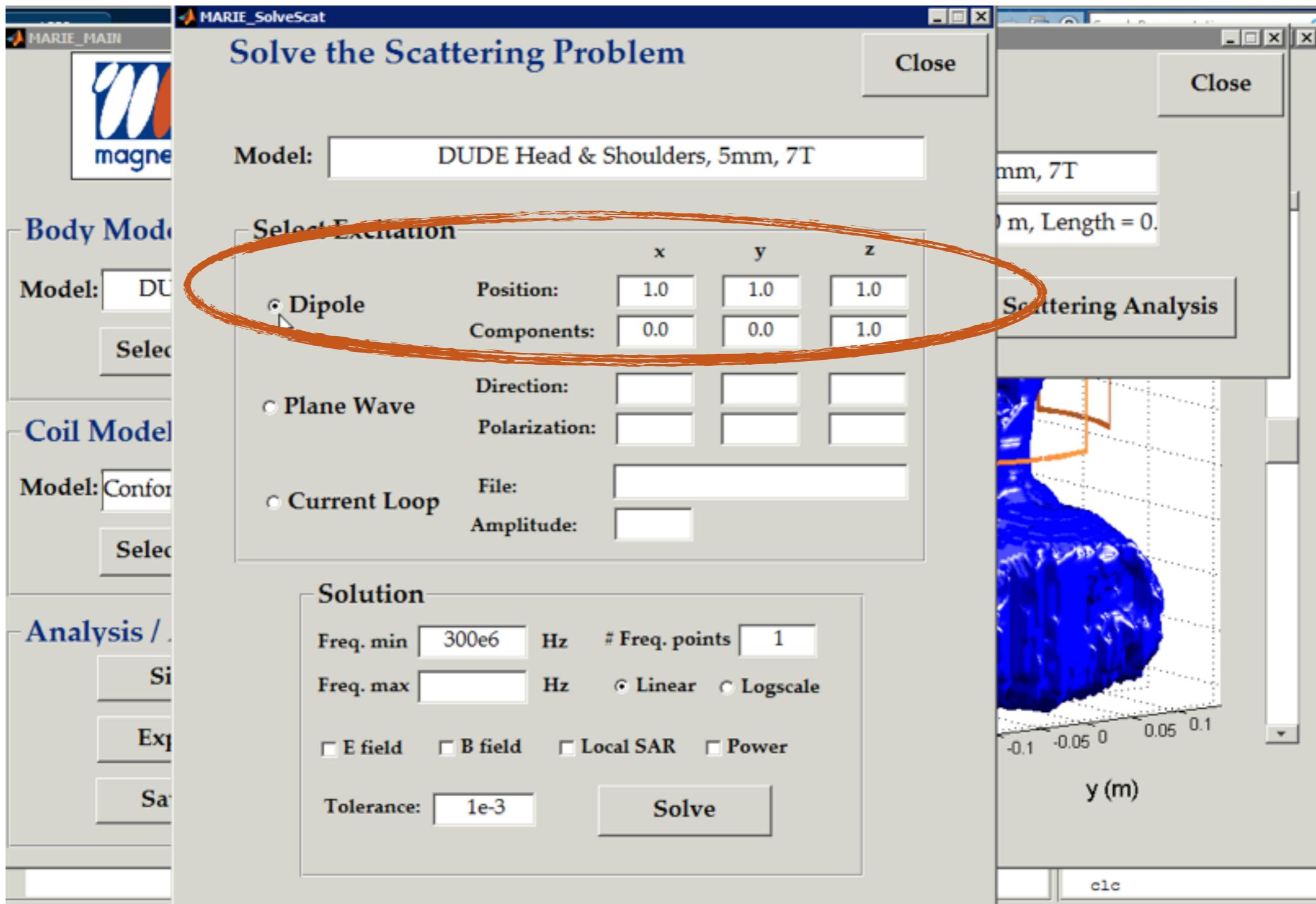


## ♦ Scattering analysis

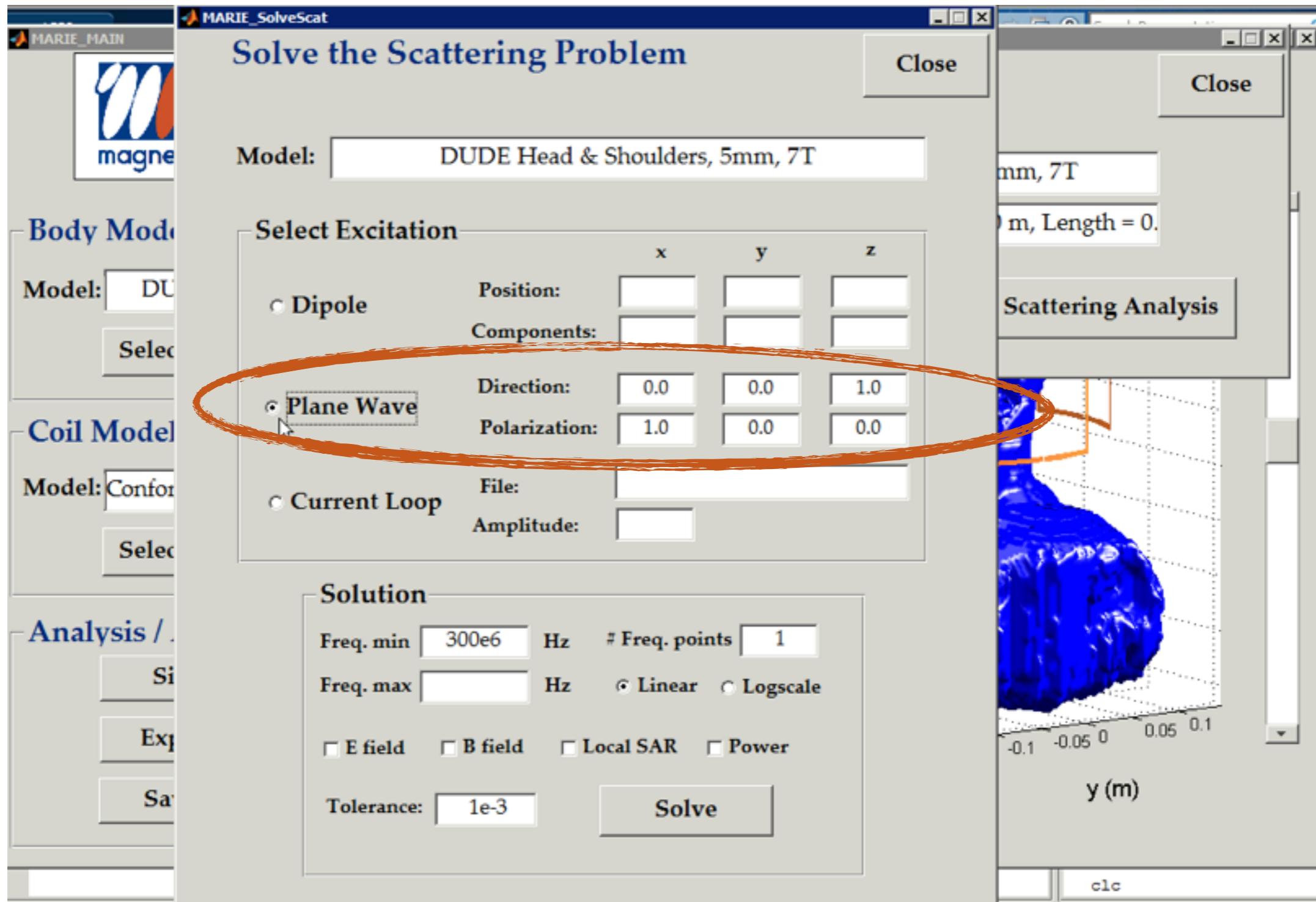
- ♦ dialogue to define excitation and frequencies opens



- ♦ Scattering analysis
  - ♦ dipole excitation: define position and components

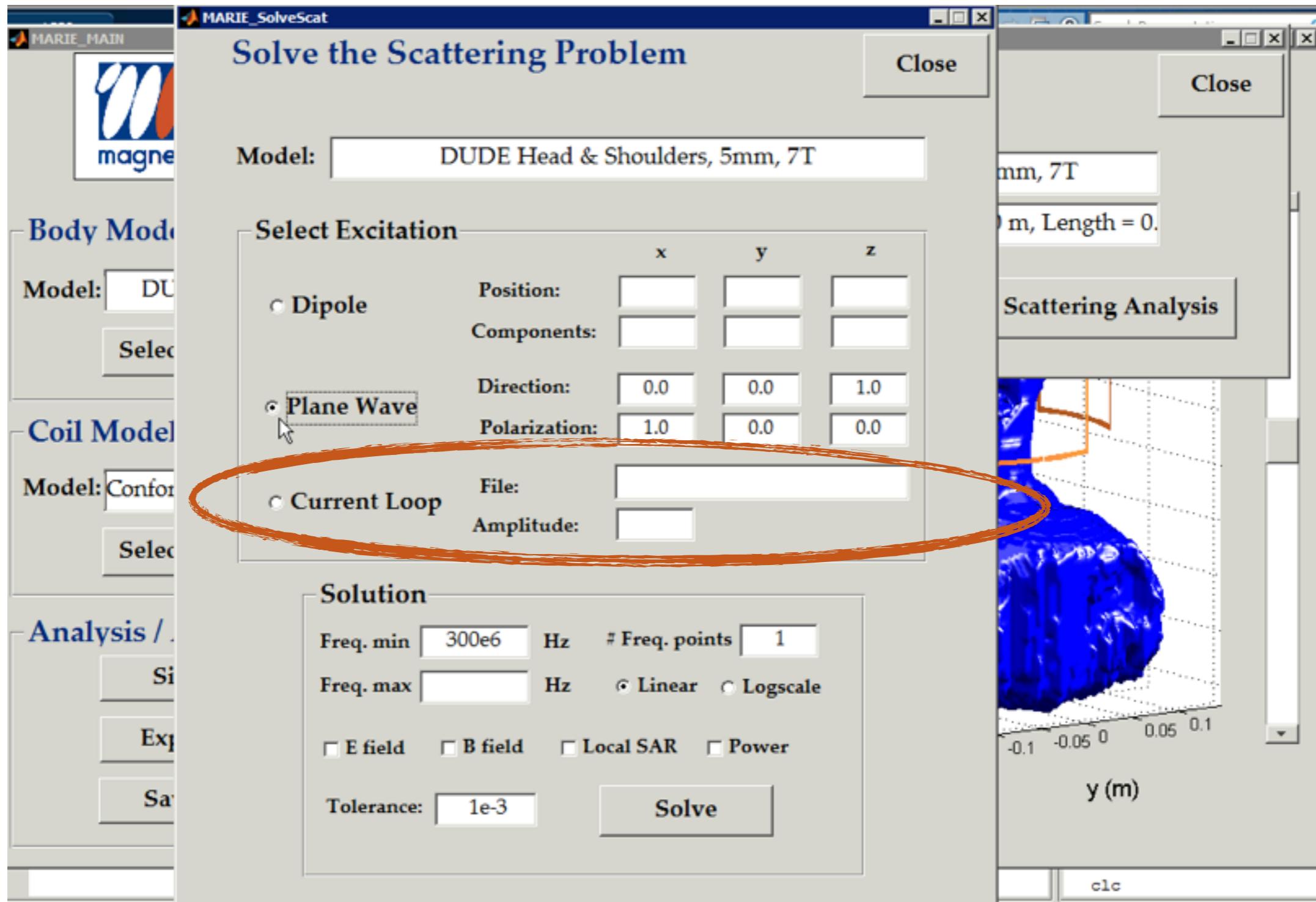


- ♦ Scattering analysis
  - ♦ plane wave excitation: define traveling direction and polarization



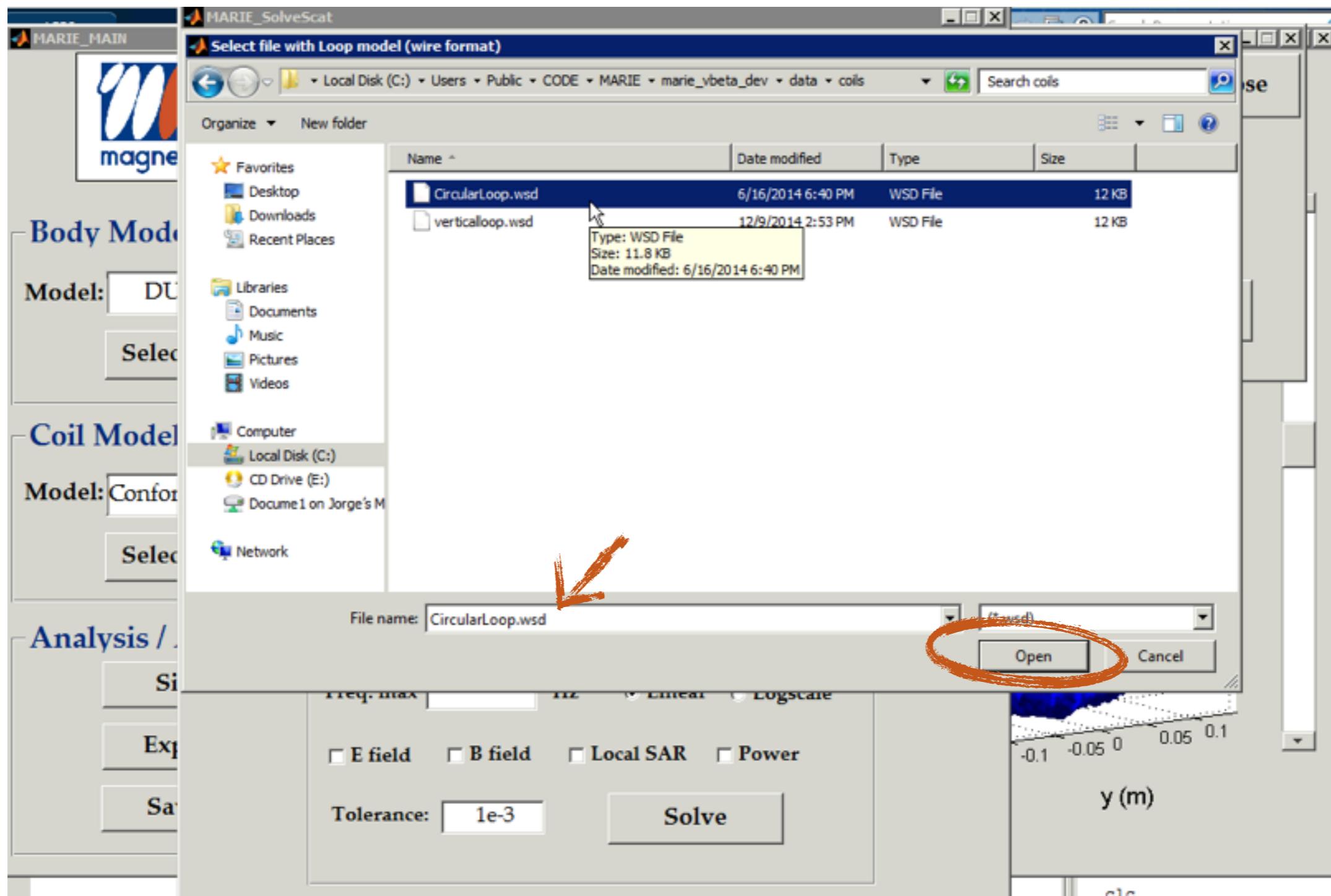
## ♦ Scattering analysis

- ♦ Constant current loop: define amplitude and load loop from file

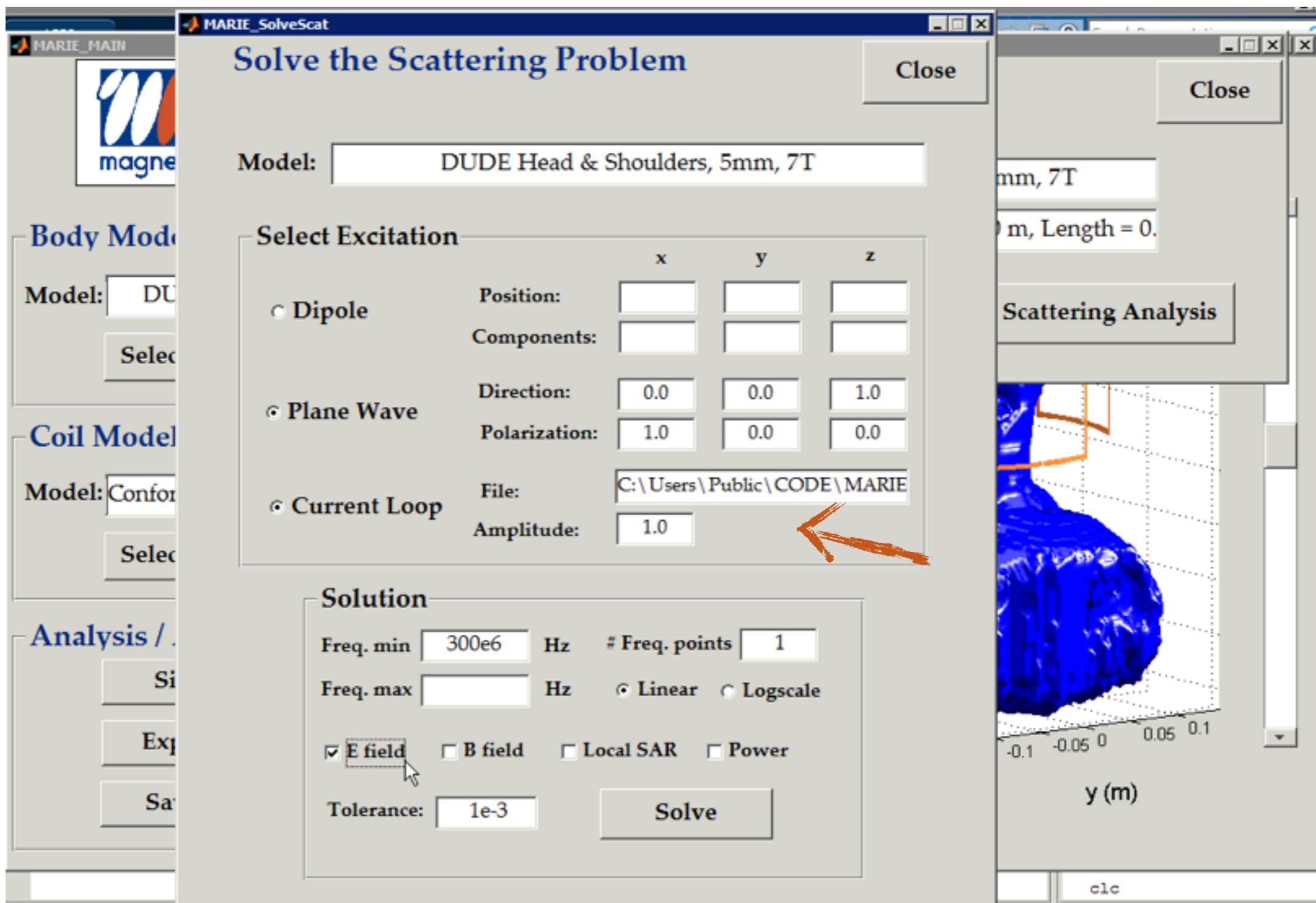


## ♦ Scattering analysis

- ♦ it will open a navigation window to select and load the file
  - ♦ it is in a .wsd file: same format as wire coil models

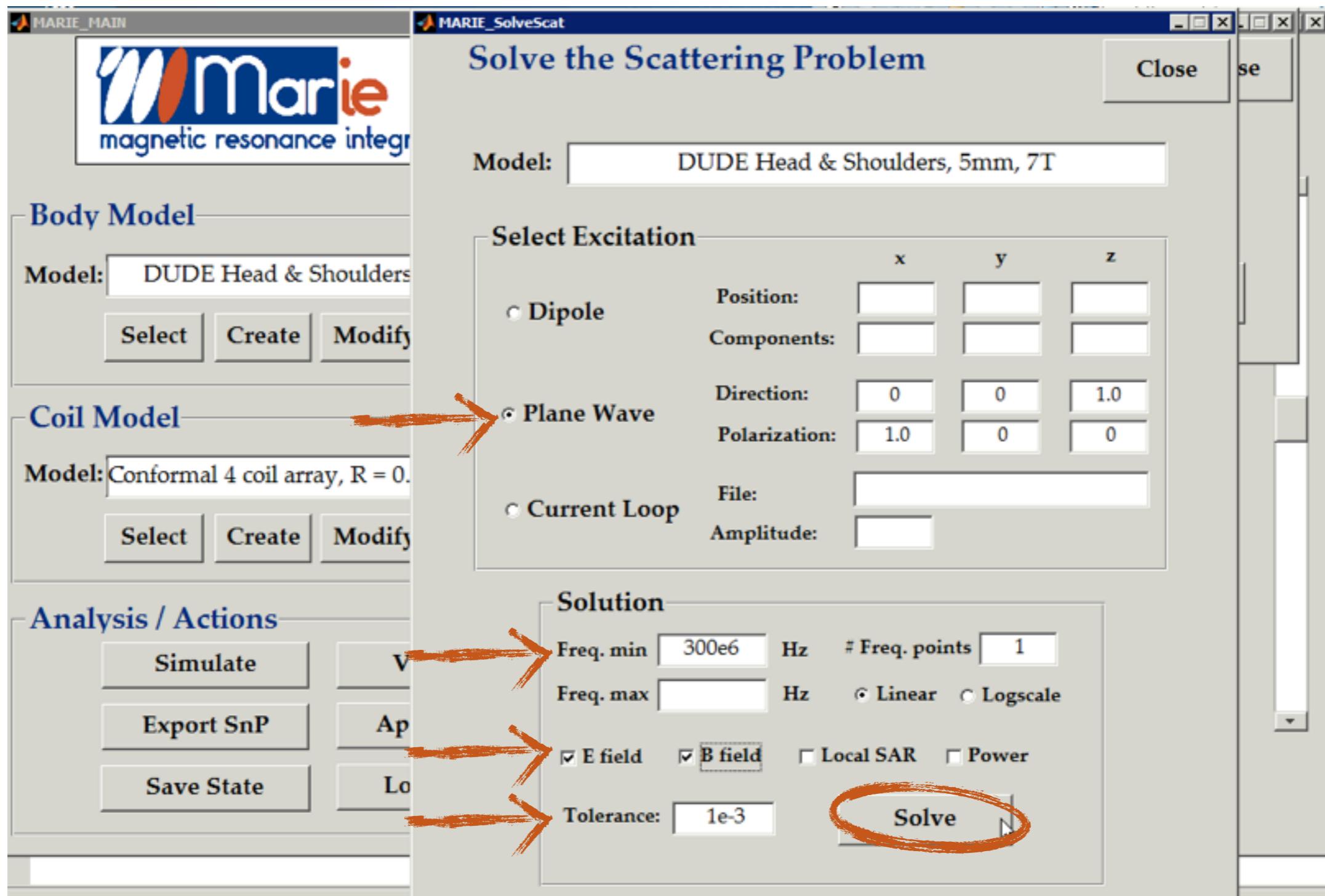


- ◆ Scattering analysis
  - ◆ it will be shown in the file box

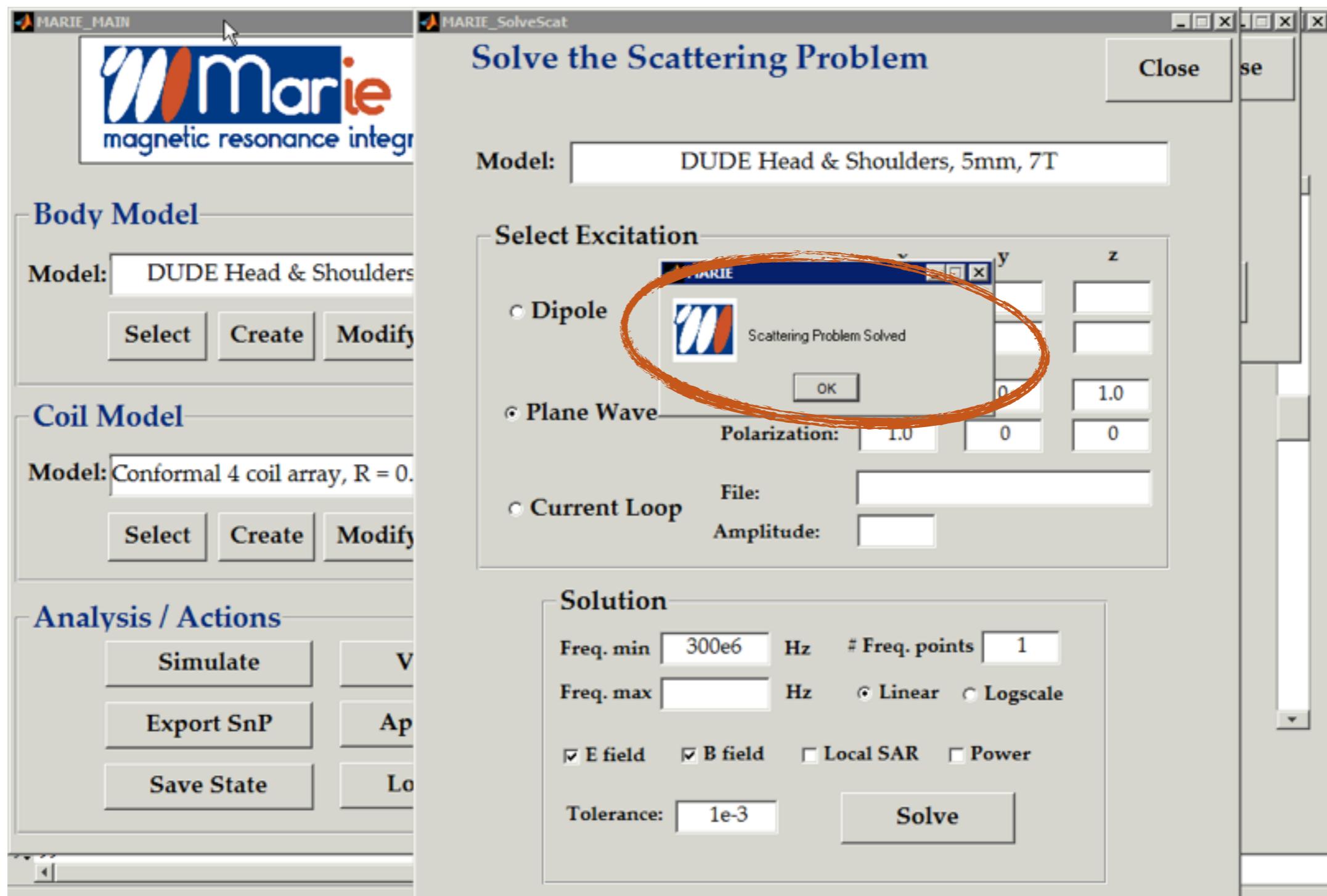


## ◆ Scattering analysis

- ◆ select analysis and error tolerance, and press solve



- ◆ Scattering analysis
  - ◆ confirmation window will show when finished



- ◆ Scattering analysis
- ◆ and simulation info on the command window

**Command Window**

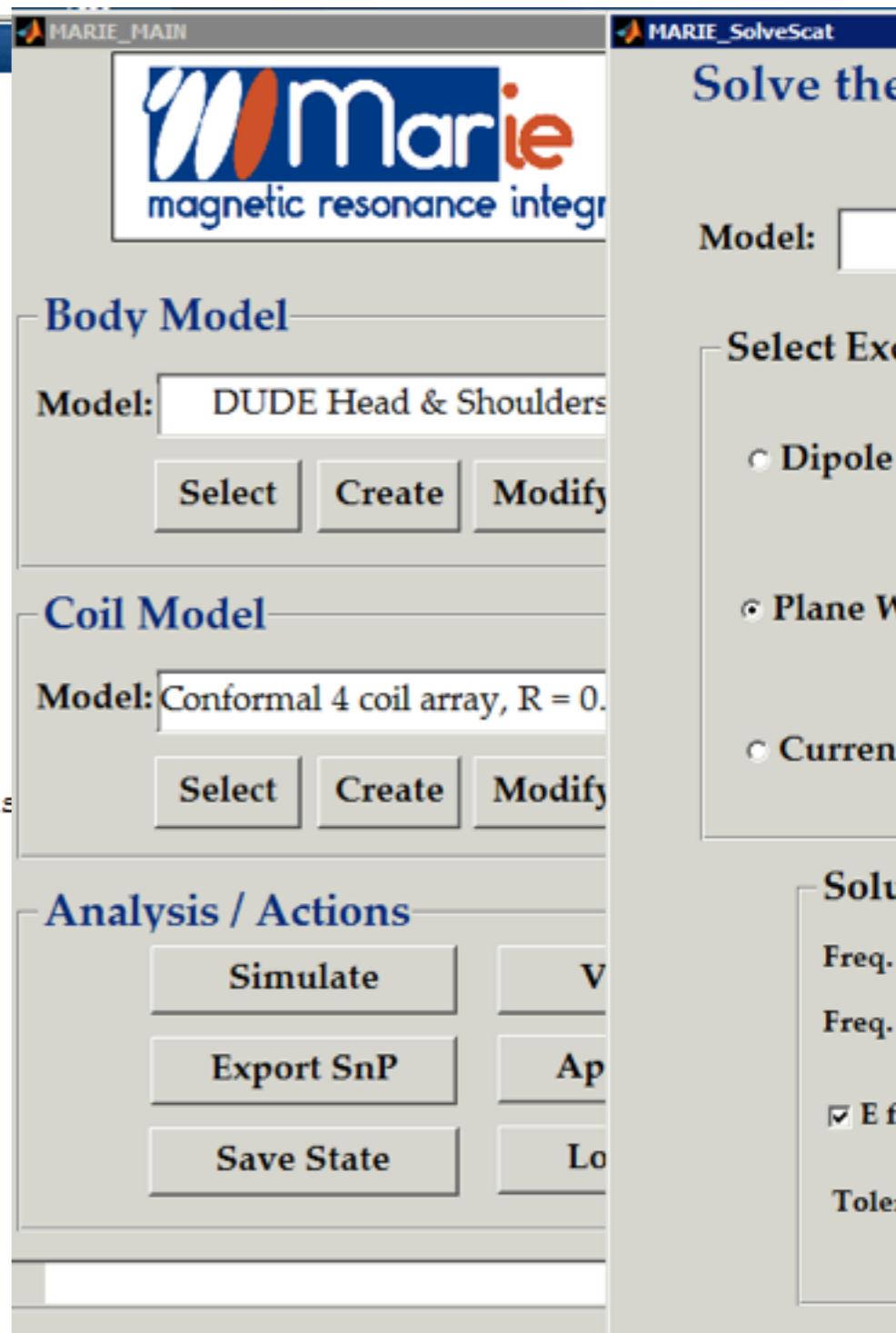
```

Domain Dimensions: 103x103x103
FFT Dimensions: 206x206x206
Memory space: 400.168579 MB
-----
Domain: 103x103x103 voxels
Resolution: 5.00mm
# DOFS: 3278181
# DOFS in Scatterer: 380574
Operating Frequency: 300.00 MHz
-----
Estimated MVP Peak Memory 3201.349 MB.

GPU BICGSTAB solver, wo/ preconditioner Time = 93.30 [sec] for 217 iterations
VIE Solve finished. Elapsed Time = 97.92 [sec]

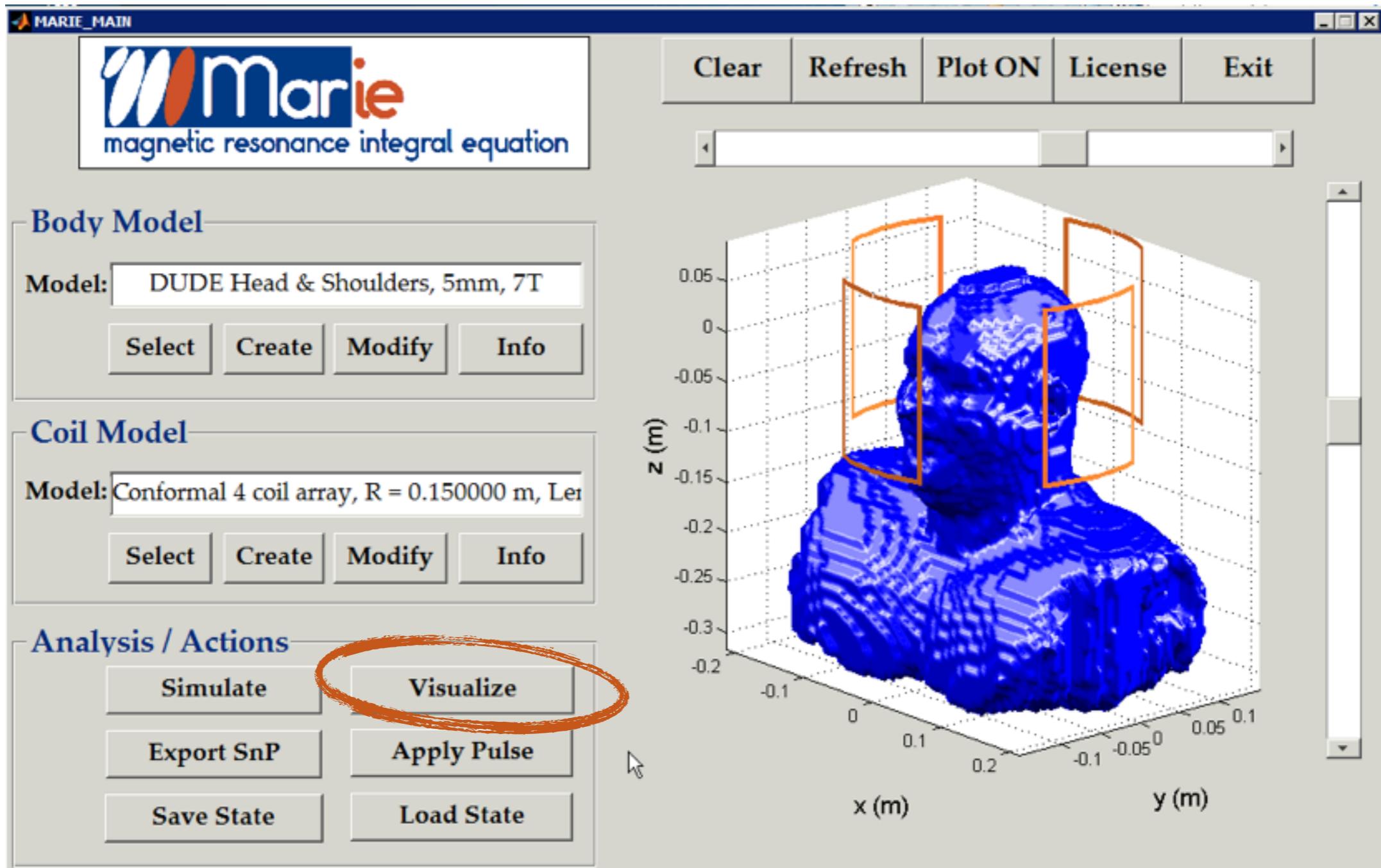
Volumetric currents generated: Elapsed time = 97.94 [sec]
SAR computed: Elapsed time = 0.15 [sec]
E fields computed: Elapsed time = 2.13 [sec]
B fields computed: Elapsed time = 1.80 [sec]
Power and GSAR computed: Elapsed time = 0.06 [sec]
-----
Overall Scattering Problem: Elapsed time = 102.10 [sec]
-----
```

fx >>



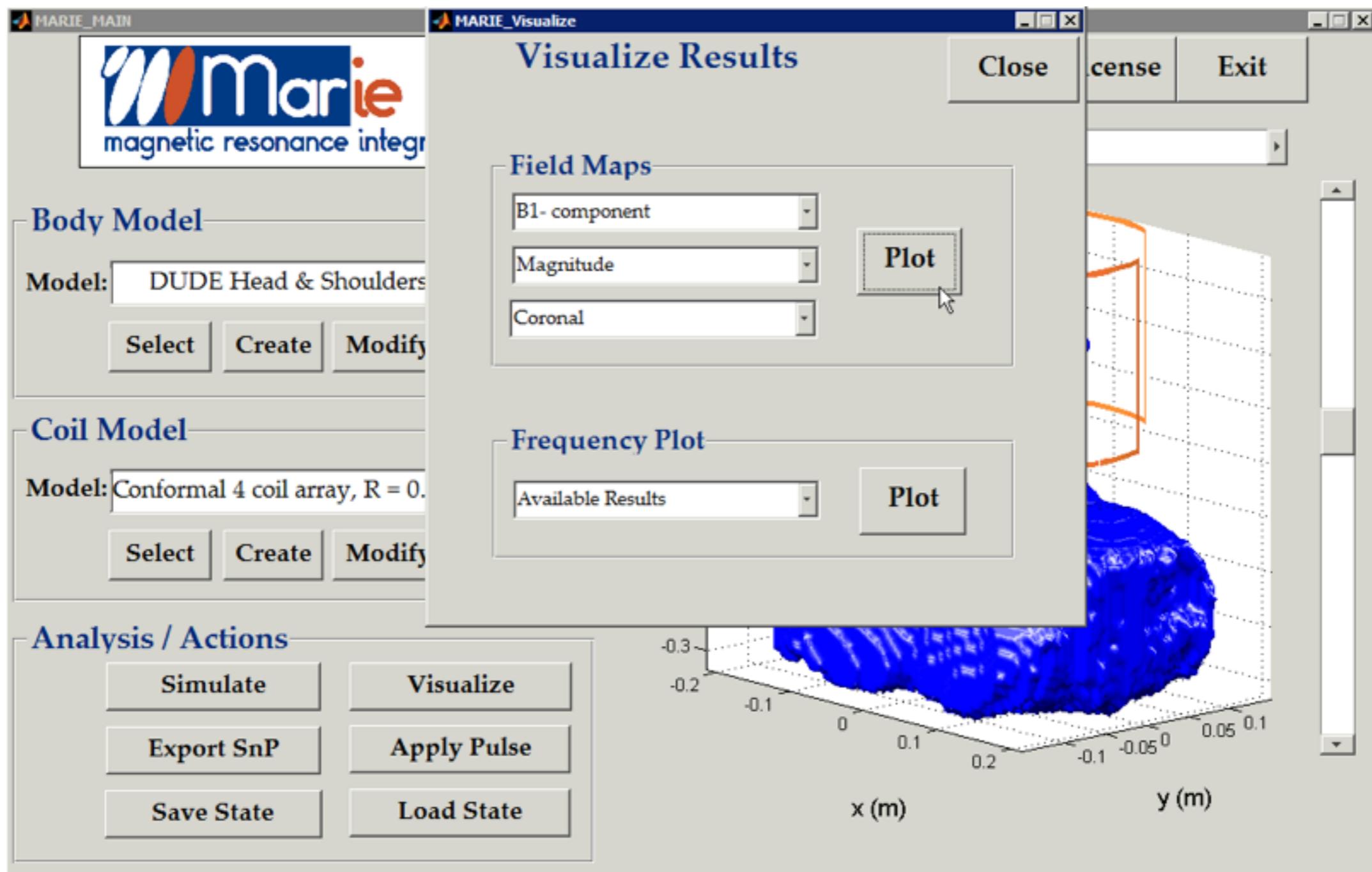
The screenshot shows the MARIE\_GUI interface. On the left, the 'Command Window' displays simulation parameters and progress. A large orange arrow points from the 'Operating Frequency' line in the command window towards the 'DUDE Head & Shoulders' model selection in the 'Body Model' section of the central panel. The central panel also includes sections for 'Coil Model' and 'Analysis / Actions'. The right panel contains tabs for 'Solve the' and 'Select Exc', along with options for 'Dipole', 'Plane W', and 'Current'.

- ◆ Visualize the results
  - ◆ push the “visualize” button



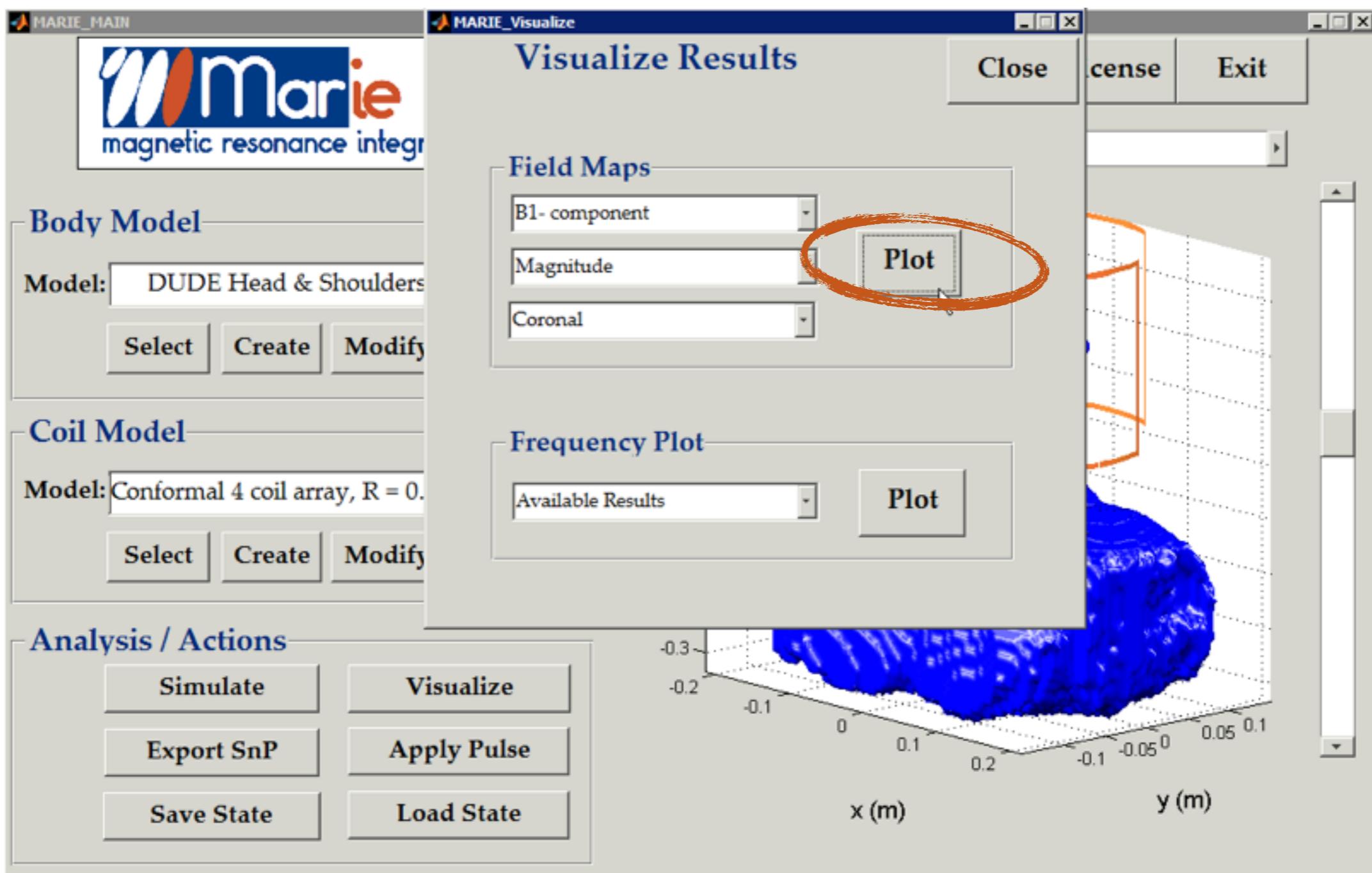
## ♦ Visualize the results

- ◆ frequency based results do not include the port parameters
- ◆ only body model related results

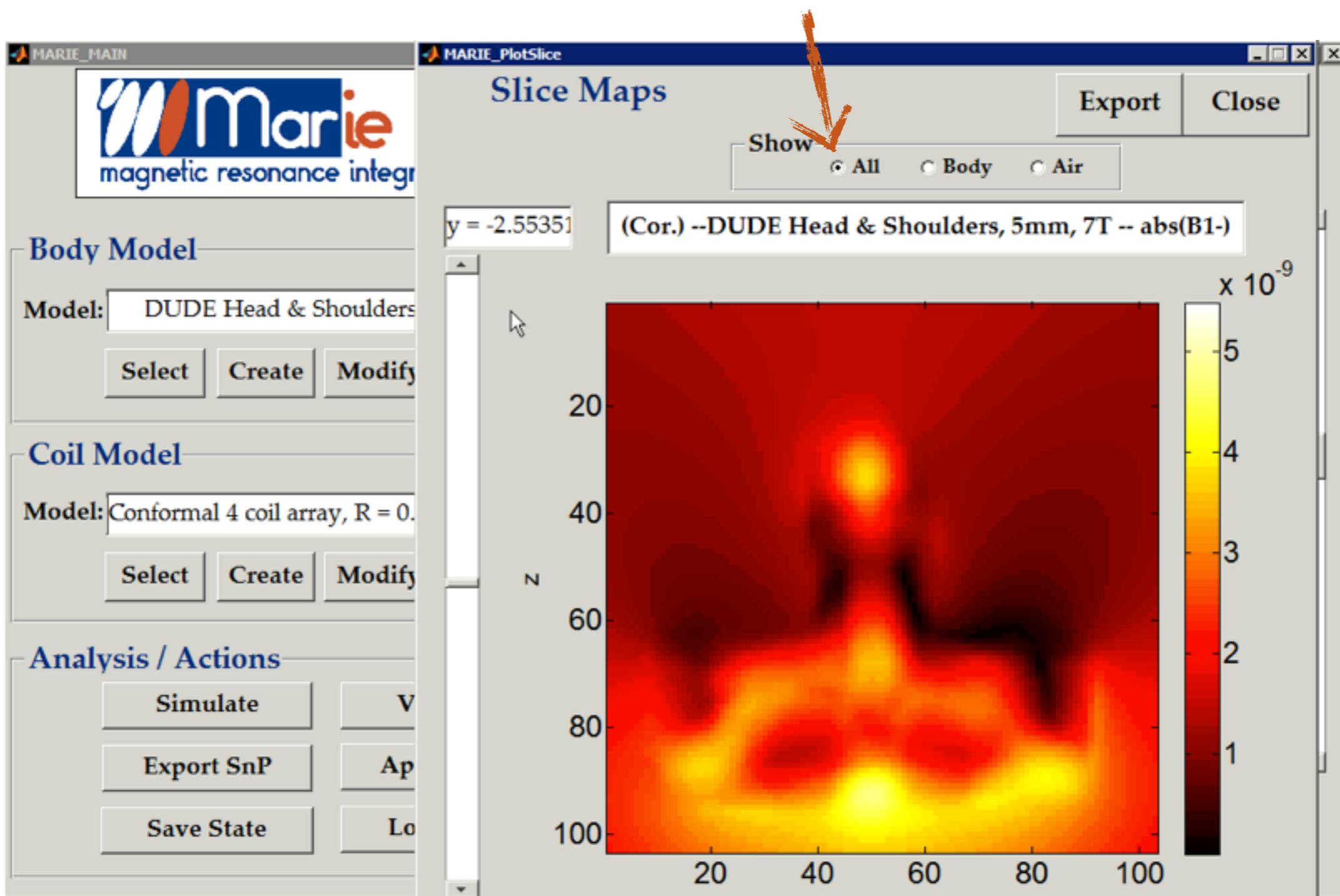


## ♦ Visualize the results

- ♦ select the desired result, map and cut
- ♦ push “plot” button to open the map figure

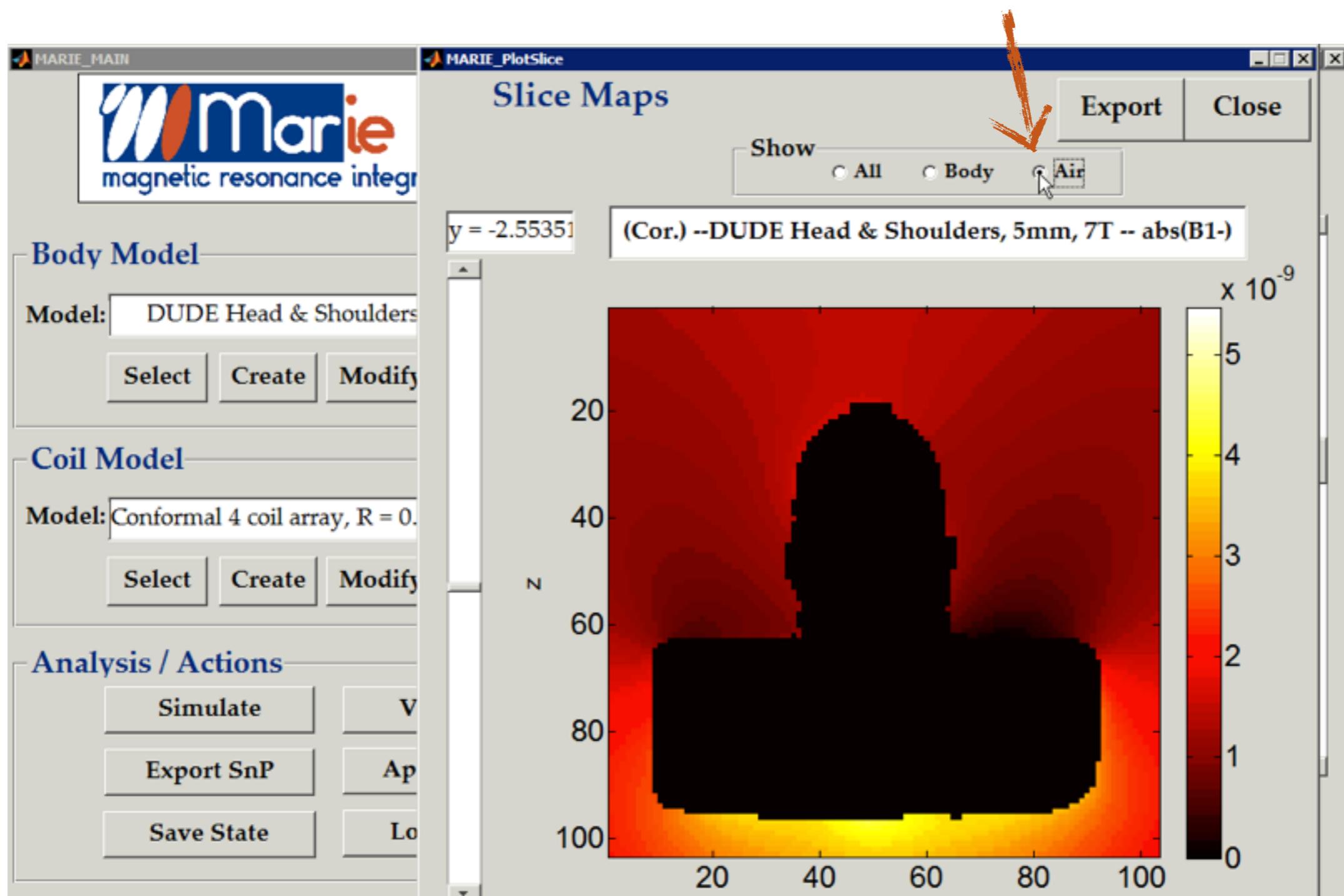


- ◆ Visualize the results
  - ◆ the slice map utility opens

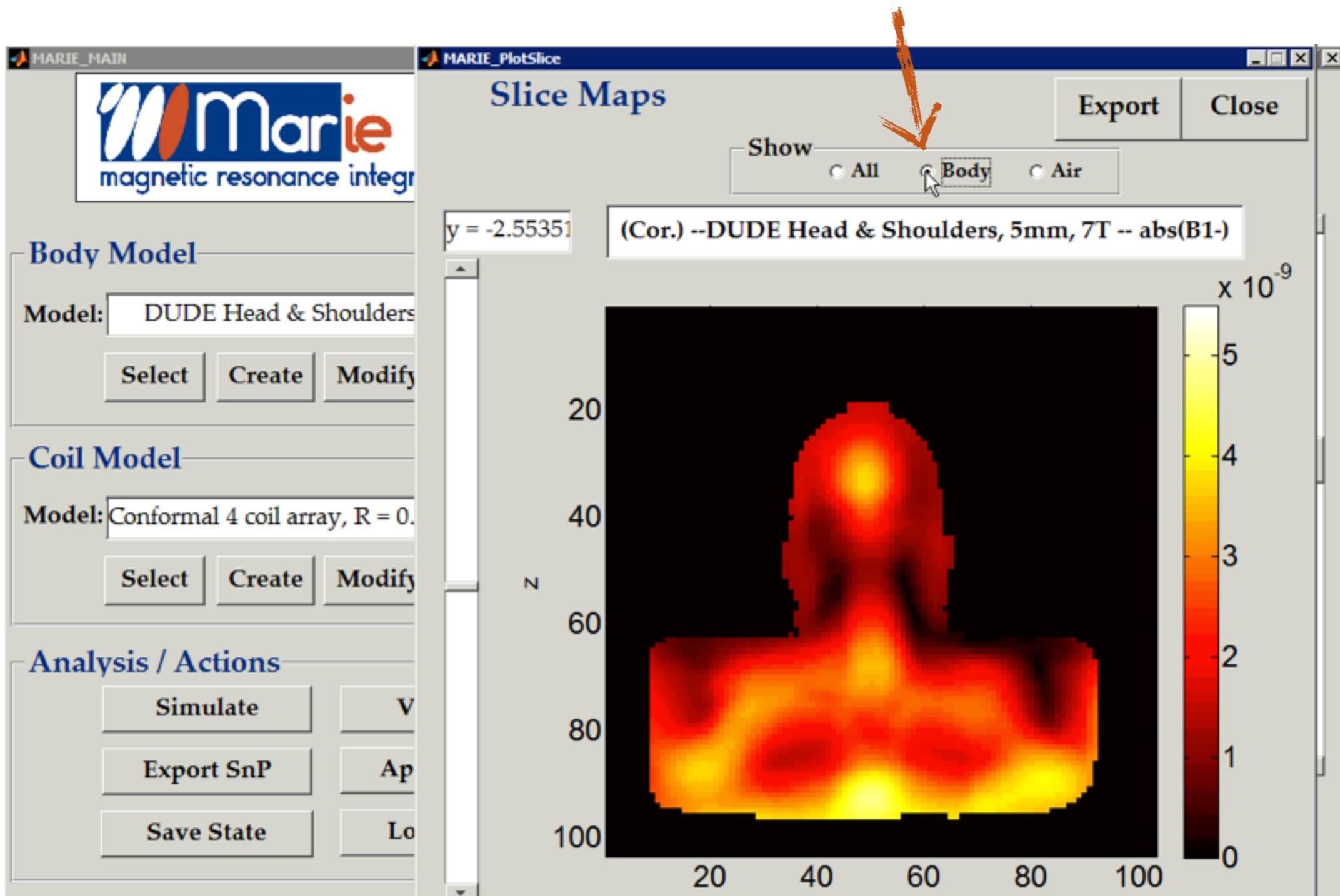


## ♦ Visualize the results

- ♦ we can mask the fields in the body to show only fields outside



- ♦ Visualize the results
  - ♦ or we can mask fields in the air for better scaling of body fields



enjoy Marie