CoilGen Documentation

Philipp Amrein

February 2022

Contents

0.1 Purpose, Introduction

This *CoilGen* project is supposed to be a community based tool that facilitates the design of coils within the MRI/NMR environment. A stream function approach of the Boundary Element Method is used to generate a current density to fit a freely specified target field. Further post processing is implement that processes the current density into a suitable coil layout. Up to now, the code is written in MATLAB, but future migration to python might be advantageous, especially since it does not need proprietary software licenses. The author is very willing to collaborate with anyone who wants do the translation.

0.2 Installation

The project requires MATLAB and additionally FastHenry2 for calculation of the inductance. The MATLAB version should not be older than *2020A*.

0.3 Code description

The algorithm is written within the main function named ”coilgen.m”. It can be called with various necessary and optional input parameters.

0.3.1 File structure

- Documentation
- Examples: Here are the Matlab scripts of the different projects that calls the main algorithm *CoilGen.m* together with the input parameters
- Geometry_Data: *.stl* files of surface geometries for the definition of the current carrying surfaces and the target fields. It should be in the bindary format.
- sub_functions: Folder of *MATLAB* sub-functions that are called by the main function *CoilGen.m* algorithm
- Pre_Optimized_Solutions: *.mat* data containers pre-optimized stream functions. Please check the function *load_preoptimized_data.m*
- plotting: Several functions to plot results.
- Results: Folder for saving results.

0.4 Usage

"Coilgen.m" must be called with the required set of input parameters. All other settings and variables are optional. If not specified the default values set. The definitions of the default values can be seen within the function *parse_inputs.m*.

0.4.1 Necessary input parameters

- *coil_mesh_file*: The name of the *.stl* file for the current carrying surface. It must be in the *Geometry_Data* folder.
- *field_shape*: The functional that specifies the target field i.e. the z component of the magnetic field i.e. B_z. In general, it can be any function of the Cartesian coordinates x, y, z. To generate a linear transverse gradient specify: $field\_shape =' y'$. This variable must be a character array.

  Other examples:

  - field_shape='x' (linear x-Gradient)
  - field_shape='y' (linear y-Gradient)
  - field_shape='z' (linear z-Gradient)
  - field_shape='$x. * sin(\alpha) - y. * cos(\alpha)$' (linear x-Gradient rotated with angle α around the z-axis)
  - field_shape='$2. * x. * y$' (Spherical harmonic $S2$ shim field)
  - field_shape='1' (constant field, $B0$ offset)

  The x, y, z coordinates refer to the coordinate system which is also used for the coordinates of the surface geometry. **Important**: Use the MATLAB $.*$ operator for element wise multiplication.

0.4.2 Optional input parameters

All other settings and variables are optional. If not specified the default values set. The definitions of the default values can be seen within the function *parse_inputs.m*.

- *pot_offset_factor*: Offset factor for contour levels, default value: 0,5, Type: numeric
- *target_mesh_file*: File of the target surface mesh, which will define the target field, default value: 'none', Type: Character array
- *secondary_target_mesh_file*: File of a secondary target surface mesh, which will define the target field, default value: 'none', Type: Character array
- *secondary_target_weight*: weight for the secondary target points , default value: 1, Type: numeric
- *use_only_target_mesh_verts*: flag to use only the target mesh vertices as target coordinates, default value: false, Type: logical
- *sf_source_file*: file of an already optimized stream function, default value: 'none', Type: Character array
- *levels*: Number of potential levels, default value: 10, Type: numeric
- *level_set_method*: Specify one of the three ways the level sets are calculated: "primary","combined", or "independent", Default value: "primary" 0,5, Type: Character array
- *fieldtype_to_evaluate*: Fieldtype to evaluate; 'gradient' or 'field', default value: 'field', Type: Character array
- *surface_is_cylinder_flag*: Fag for cylindrical surface; in case of cylinder, a special parameterization will be used: Default value: true, Type: Logical
- *circular_diameter_factor_cylinder_parameterization*: for the cylinder parameterization the ration of outer and inner boundary, default value: 1, Type: numeric
- *interconnection_cut_width*: The width in meter of the opening cut for the interconnection of the loops: Default value: 0.01, Type: numeric
- *target_region_radius*: Radius of a spherical target field, default value: 0.15, Type: numeric

- *target_region_resolution*: Number of target points per dimension within the target region: Default value: 10, Type: numeric
- *normal_shift_length*: The distance in meter for which crossing lines will be seperated along the normal direction of the surface, Default value: 0.001, Type: numeric
- *min_point_loop_number*: The minimal required number of point of a single loop; otherwise loops will be removed.., default value: 20, Type: numeric
- *area_perimeter_deletion_ratio*: Additional loop removal criteria which relates to the perimeter to surface ratio of the loop, default value: 5, Type: numeric
- *max_allowed_angle_within_coil_track*: Max allowed angle of the track of the contours, Default value: 120, Type: numeric
- *min_allowed_angle_within_coil_track*: Min allowed angle of the track of the contours; smaller angles will be converted to straight lines in order to reduce the number of points, Default value: 0.0001, Type: numeric
- *tiny_segment_length_percentage*: Minimum relative percentage for which points will be deleted which contribute to segments which is extremly short, default value: 0, Type: numeric
- *iteration_num_stream_func_refinement*: Number of refinement iterations of the mesh (together with the stream function), default value: 0, Type: numeric
- *b_0_direction*: The direction (vector) along the interconnections will be aligned, default value: [0 0 1], Type: numeric array
- *track_width_factor*: In case of pcb layout, specify the track width: 0,5, Type: numeric
- *conductor_cross_section_width*: cross section width of the conductor (for the inductance calculation) in meter, default value: 0.002, Type: numeric
- *conductor_cross_section_height*: cross section height of the conductor (for the inductance calculation) in meter, default value: 0.002, Type: numeric
- *specific_conductivity_conductor*: Conducter conductiviy, default value:$0.018 * 10^{-6}$(Copper), Type: numeric

- *conductor_thickness*: Thickness of the sheet current density of within the stream function representation: 0.005, Type: numeric
- *cross_sectional_points*: 2D edge points for direct defintion of the cross section of the conductor, default value: $[21.5; 2-1.5; -2-1.5; -21.5; 21.5]'/2000$, Type: numeric array
- *geometry_source_path*: Directory of the .stl geometry files: 'Geometry_Data', Type: Character array
- *output_directory*: Output directory for saving results, default value: 'Results', Type: character array
- *save_stl_flag*: Flag to save sweeped .stl: false, Type: logical
- *plot_flag*: Flag to plot results, default value: true, Type: logical
- *interconnection_method*: Interconnection method: Regular or spiral in/out, default value: 'regular', Type: Character array
- *skip_postprocessing*: Flag to skip post processin, default value: false, Type: logical
- *skip_inductance_calculation*: Flag to skip inductance calculation, default value: false, Type: logical
- *force_cut_selection*: Force selection of the orientation of the interconnection areas {'low' 'high'...}, default value: {}, Type: cell
- *gauss_order*: Gaus integration order, for the calculation of the sensitivity matrix default value: 2, Type: numeric
- *set_roi_into_mesh_center*: Flag to set the roi into the geometric center of the mesh, default value:false, Type: logical
- *tikonov_reg_factor*: Tikonov regularization factor for the SF optimization, default value: 1, Type: numeric

0.5 How to design your own coil with CoilGen

Make a copy of *y_gradient_coil.m* within the folder *Examples* as a starting point.

Wihtin this script, adjust the input parameters of the call of CoilGen to your need. Especially, make sure the following parameters are well specified:

- field_shape: Choose the kind of target field: 'x' ,'y' ,'z' etc.
- coil_mesh_file: Specify the .stl file of the current carrying surface
- levels: Specifiy the number of turns of the coil
- surface_is_cylinder_flag: This relates to the surface parameterization. If your current carrying is similar to a cylinder, set this flag to true
- interconnection_cut_width: The width of the connections, which determines their steepness
- normal_shift_length: The distance in meter for which crossing lines will be seperated along the normal direction of the surface
- iteration_num_stream_func_refinement: Choose a number between (0,1,2). The higher the number, the smoother the result, but the greater the computational effort.