EMDLAB finite element package

Electrical Machines Design Laboratory

Overview:

EMDLAB is an open-source numerical package developed in the MATLAB environment for the design and analysis of electrical machines, including motors, generators, transformers, and actuators. The package provides a transparent and reproducible tool for academic research and education. Its open-source nature encourages collaboration and facilitates easy extension of the code, making it particularly valuable for method development and academic use.

The package is organized as a library of MATLAB objects. Depending on your specific design needs, you can select the appropriate modules to obtain your desired results. With EMDLAB, you can also develop customized, standalone software tailored to your applications.

Setup Instructions (Windows 64-bit):

1. Download the emdlab-win64.zip file.
2. Extract the zip file and place the "emdlab-win64" folder in the "C:" directory, without changing the folder name.
3. To use the EMDLAB package in your MATLAB code, add the following line at the beginning of your mfile:
   --->> addpath(genpath('C:\emdlab-win64'));

How to install EMDLAB from GitHub? (Follow the video link):

https://github.com/EMDLAB-Package/emdlab-win64

https://youtu.be/ifwybm4r2_0

For academic use, cite this paper:
https://www.sciencedirect.com/science/article/pii/S2352711025004121

How to use Gmsh as a mesh generator in EMDLAB:

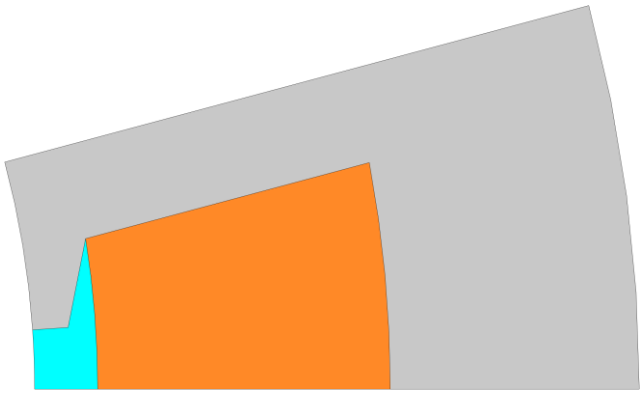
In addition to the built-in EMDLAB mesh generators, it is also possible to use Gmsh as an external mesh generator.

1. Install Python.
2. Install Gmsh -> you can use: pip install gmsh
3. Set python.exe path in the pyPath.txt file located in "C:\emdlab-win64\geometry\pyPath.txt"
4. When using the .generateMesh method, use 'gmsh' as input: .generateMesh('gmsh')

Geometry Library

In this section, the functions available in the geometry library of EMDLAB are explained.

emdlab_g2d_lib_tc1



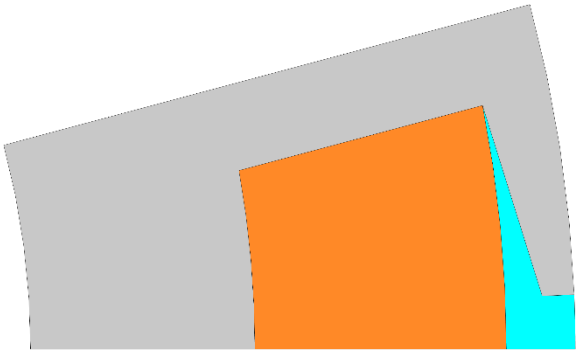
Inputs:

1. g: geometry database handle: emdlab_g2d_db
2. ID: inner diameter: a double
3. OD: outer diameter: a double
4. Ns: number of slots: an integer
5. wt: tooth width: a double
6. dss: depth of stator slot: a double
7. bs0: slot opening: a double
8. hs0: slot opening height: a double
9. tta: tooth tip angle: in degrees
10. name1: a name for iron region, for example: “stator”: must be char or string.
11. name2: a name for coil arm region, for example: “sca”: must be char or string.
12. name3: a name for slot air region, for example: “sap”: must be char or string.

Outputs:

After running this function all related geometrical entities will be added to the geometry database.

emdlab_g2d_lib_tc2



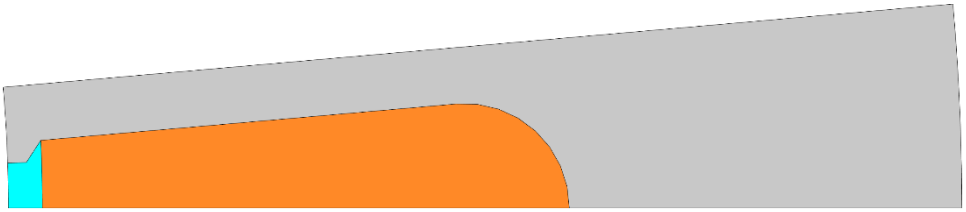
Inputs:

1. g: geometry database handle: emdlab_g2d_db
2. ID: inner diameter: a double
3. OD: outer diameter: a double
4. Ns: number of slots: an integer
5. wt: tooth width: a double
6. dss: depth of stator slot: a double
7. bs0: slot opening: a double
8. hs0: slot opening height: a double
9. tta: tooth tip angle: in degrees
10. name1: a name for iron region, for example: “stator”: must be char or string.
11. name2: a name for coil arm region, for example: “sca”: must be char or string.
12. name3: a name for slot air region, for example: “sap”: must be char or string.

Outputs:

After running this function all related geometrical entities will be added to the geometry database.

emdlab_g2d_lib_tc3



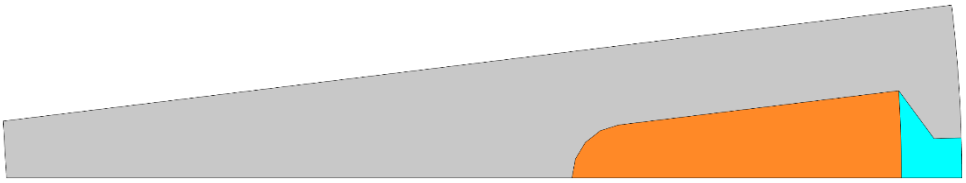
Inputs:

1. g: geometry database handle: emdlab_g2d_db
2. ID: inner diameter: a double
3. OD: outer diameter: a double
4. Ns: number of slots: an integer
5. wt: tooth width: a double
6. dss: depth of stator slot: a double
7. bs0: slot opening: a double
8. hs0: slot opening height: a double
9. tta: tooth tip angle: in degrees
10. name1: a name for iron region, for example: “stator”: must be char or string.
11. name2: a name for coil arm region, for example: “sca”: must be char or string.
12. name3: a name for slot air region, for example: “sap”: must be char or string.

Outputs:

After running this function all related geometrical entities will be added to the geometry database.

emdlab_g2d_lib_tc4



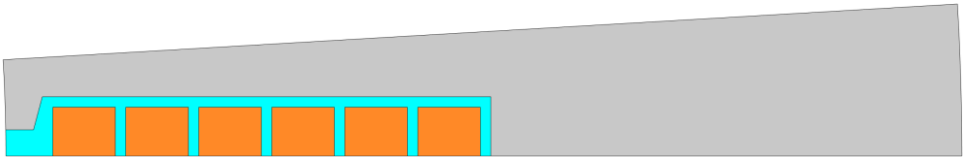
Inputs:

1. g: geometry database handle: emdlab_g2d_db
2. ID: inner diameter: a double
3. OD: outer diameter: a double
4. Ns: number of slots: an integer
5. wt: tooth width: a double
6. dss: depth of stator slot: a double
7. bs0: slot opening: a double
8. hs0: slot opening height: a double
9. tta: tooth tip angle: in degrees
10. name1: a name for iron region, for example: “stator”: must be char or string.
11. name2: a name for coil arm region, for example: “sca”: must be char or string.
12. name3: a name for slot air region, for example: “sap”: must be char or string.

Outputs:

After running this function all related geometrical entities will be added to the geometry database.

emdlab_g2d_lib_tc9



Inputs:

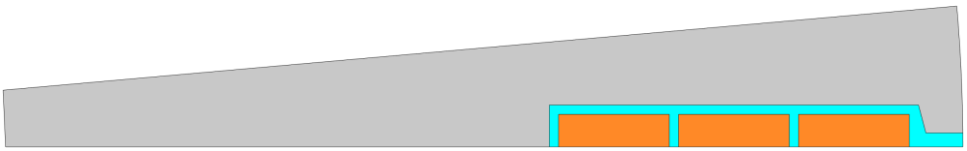
1. g: geometry database handle: emdlab_g2d_db
2. ID: inner diameter: a double
3. OD: outer diameter: a double
4. Ns: number of slots: an integer
5. Nc: number of conductors in each slot: an integer
6. wc: width of a conductor: a double
7. dc: depth of a conductor: a double
8. bs0: slot opening: a double
9. hs0: slot opening height: a double
10. tta: tooth tip angle: in degrees
11. d0x: distance between two conductors: a double
12. d0y: distance between one conductor and iron: a double
13. name1: a name for iron region, for example: “stator”: must be char or string.

14. name2: a name for coil arm region, for example: “sca”: must be char or string.
15. name3: a name for slot air region, for example: “sap”: must be char or string.

Outputs:

After running this function all related geometrical entities will be added to the geometry database.

emdlab_g2d_lib_tc10



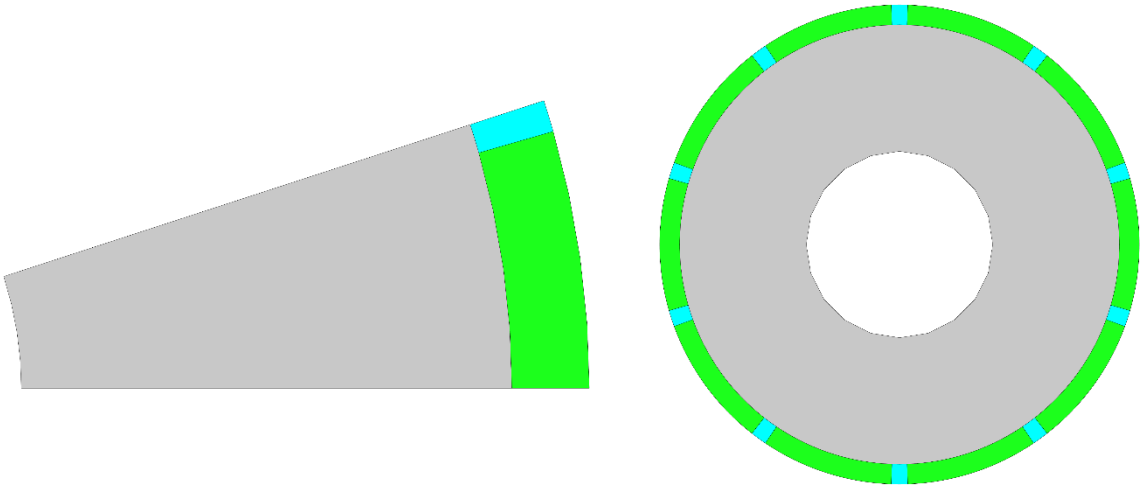
Inputs:

1. g: geometry database handle: emdlab_g2d_db
2. ID: inner diameter: a double
3. OD: outer diameter: a double
4. Ns: number of slots: an integer
5. Nc: number of conductors in each slot: an integer
6. wc: width of a conductor: a double
7. dc: depth of a conductor: a double
8. bs0: slot opening: a double
9. hs0: slot opening height: a double
10. tta: tooth tip angle: in degrees
11. d0x: distance between two conductors: a double
12. d0y: distance between one conductor and iron: a double
13. name1: a name for iron region, for example: “stator”: must be char or string.
14. name2: a name for coil arm region, for example: “sca”: must be char or string.
15. name3: a name for slot air region, for example: “sap”: must be char or string.

Outputs:

After running this function all related geometrical entities will be added to the geometry database.

emdlab_g2d_lib_rm_spm1



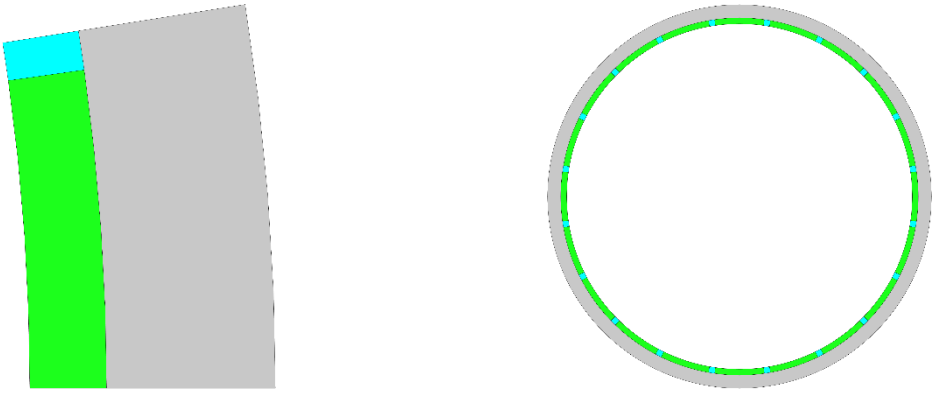
Inputs:

1. g: geometry database handle: emdlab_g2d_db
2. ID: inner diameter: a double
3. OD: outer diameter: a double
4. p: number of poles: an integer
5. dm: magnet depth: a double
6. embrace: magnet embrace ratio: a double: must be lower than 1
7. name1: a name for iron region, for example: “rotor”: must be char or string.
8. name2: a name for coil arm region, for example: “magnet”: must be char or string.
9. name3: a name for slot air region, for example: “rap”: must be char or string.

Outputs:

After running this function all related geometrical entities will be added to the geometry database.

emdlab_g2d_lib_rm_spm2



Inputs:

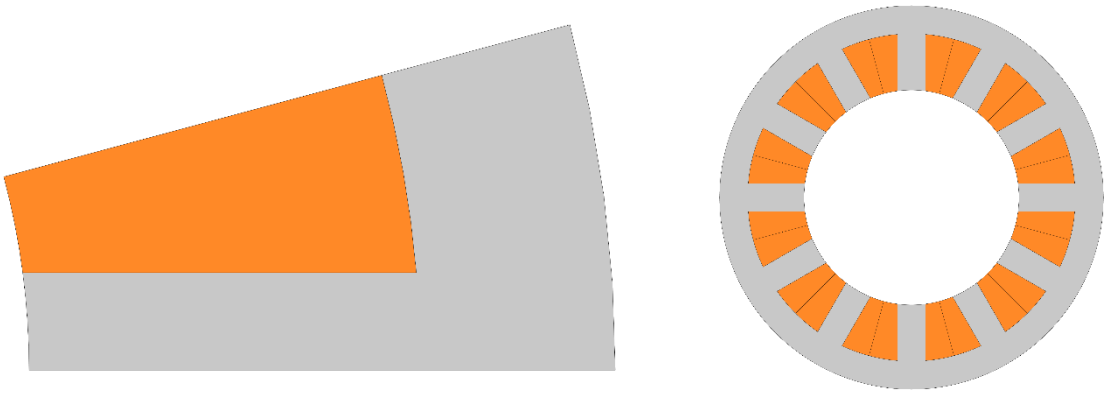
1. g: geometry database handle: emdlab_g2d_db
2. ID: inner diameter: a double
3. OD: outer diameter: a double
4. p: number of poles: an integer

5. dm: magnet depth: a double
6. embrace: magnet embrace ratio: a double: must be lower than 1
7. name1: a name for iron region, for example: “rotor”: must be char or string.
8. name2: a name for coil arm region, for example: “magnet”: must be char or string.
9. name3: a name for slot air region, for example: “rap”: must be char or string.

Outputs:

After running this function all related geometrical entities will be added to the geometry database.

emdlab_g2d_lib_tc_srm1



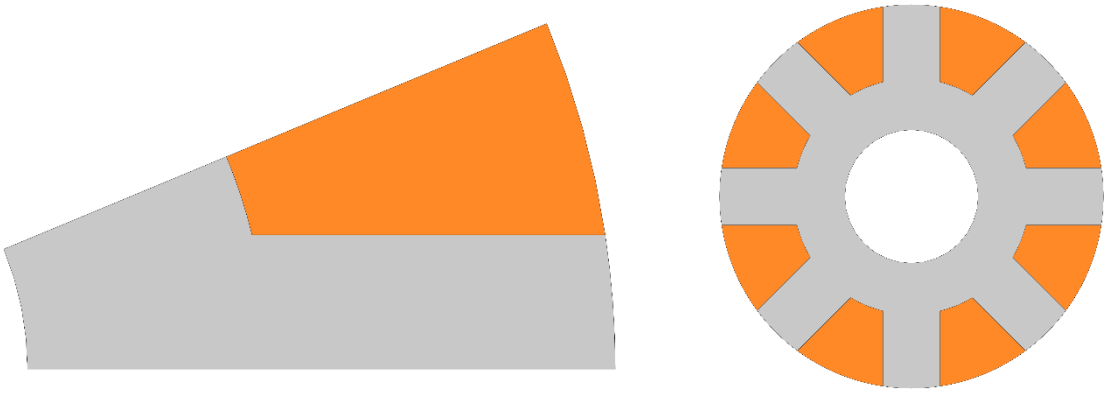
Inputs:

1. g: geometry database handle: emdlab_g2d_db
2. ID: inner diameter: a double
3. OD: outer diameter: a double
4. Ns: number of slots: an integer
5. g_beta: pole arc angle per pole pith angle: a number lower than 1
6. g_ws: yoke width per tooth width: a number lower than 1
7. name1: a name for iron region, for example: “stator”: must be char or string.
8. name2: a name for coil arm region, for example: “coil”: must be char or string.

Outputs:

After running this function all related geometrical entities will be added to the geometry database.

emdlab_g2d_lib_tc_srm2



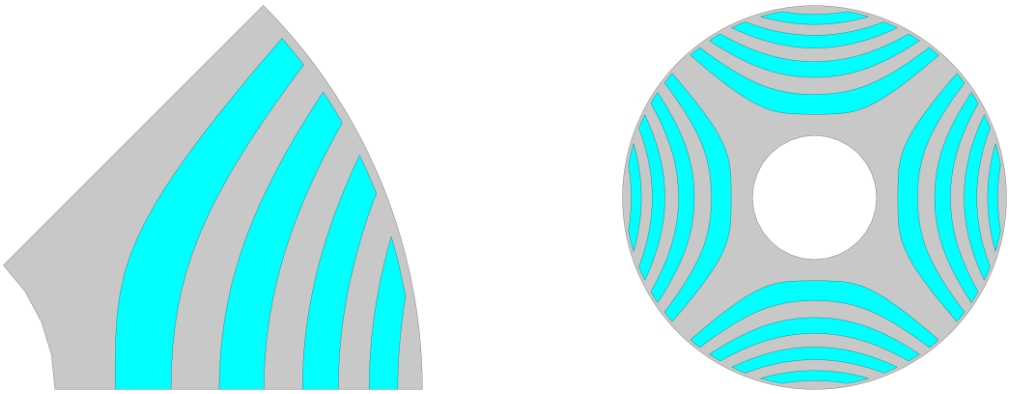
Inputs:

1. g: geometry database handle: emdlab_g2d_db
2. ID: inner diameter: a double
3. OD: outer diameter: a double
4. Ns: number of slots: an integer
5. g_beta: pole arc angle per pole pith angle: a number lower than 1
6. g_ws: yoke width per tooth width: a number lower than 1
7. name1: a name for iron region, for example: “rotor”: must be char or string.
8. name2: a name for coil arm region, for example: “rap”: must be char or string.

Outputs:

After running this function all related geometrical entities will be added to the geometry database.

emdlab_g2d_lib_rm_synrm1



Inputs:

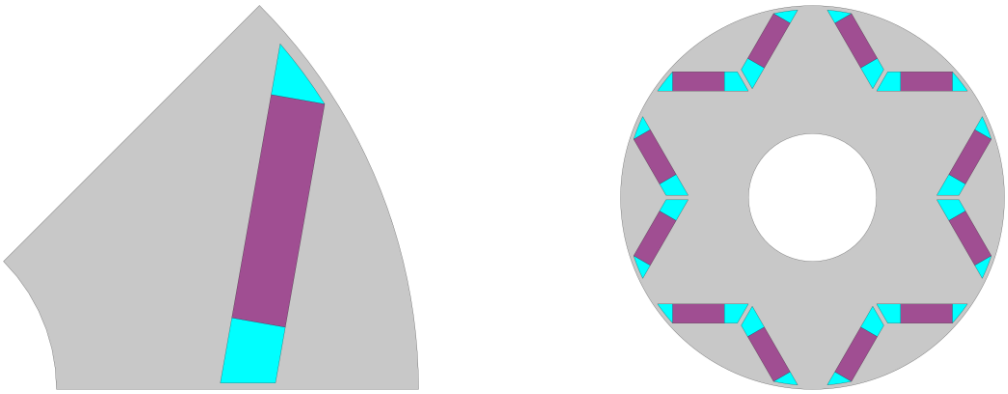
1. g: geometry database handle: emdlab_g2d_db
2. ID: inner diameter: a double
3. OD: outer diameter: a double
4. p: number of poles: an integer
5. K_air: air insulation ratio: a number lower than 1
6. g_wy: an array of numbers lower than 1
7. g_db: an array of numbers lower than 1

8. name1: a name for iron region, for example: “rotor”: must be char or string.
9. name2: a name for coil arm region, for example: “fb”: must be char or string.

Outputs:

After running this function all related geometrical entities will be added to the geometry database.

emdlab_g2d_lib_rm_ipm8



Inputs:

1. g: geometry database handle: emdlab_g2d_db
2. ID: inner diameter: a double
3. OD: outer diameter: a double
4. p: number of poles: an integer
5. alpha: ratio of v embrace to pole pitch angle: a double lower than 1
6. beta: the angle of magnet with respect to y-axis: in degrees
7. dm: depth of the magnet
8. gamma: fill factor of the magnet: a double lower than 1
9. wtrib: width of tangential rib: a double
10. wrrib: width of radial rib: a double
11. name1: a name for iron region, for example: “rotor”: must be char or string.
12. name2: a name for magnet region, for example: “magnet”: must be char or string.
13. name3: a name for air pocket regions, for example: “rap”: must be char or string.

Outputs:

After running this function all related geometrical entities will be added to the geometry database.

EMDLAB Solvers

Magnetic Static

emdlab_solvers_ms2d_tl3_ihnl

This is a two-dimensional magnetic static solver in the xy-plane with following properties:

- first order triangular mesh
- triangular lagrangian elements: 3 points per element

- isotropic
- homogenous
- nonlinear solver: Newton-Raphson

emdlab_solvers_ms2d_tl6_ihnl

This is a two-dimensional magnetic static solver in the xy-plane with following properties:

- second order triangular mesh
- triangular lagrangian elements: 6 points per element
- isotropic
- homogenous
- nonlinear solver: Newton-Raphson

Magnetic Transinet

emdlab_solvers_mt2d_tl3_ihnl_wtm

This is a two-dimensional magnetic transient solver in the xy-plane with following properties:

- first order triangular mesh
- triangular lagrangian elements: 3 points per element
- isotropic
- homogenous
- nonlinear solver: Newton-Raphson
- without motion modeling

emdlab_solvers_mt2d_tl3_ihnl_wm

This is a two-dimensional magnetic transient solver in the xy-plane with following properties:

- first order triangular mesh
- triangular lagrangian elements: 3 points per element
- isotropic
- homogenous
- nonlinear solver: Newton-Raphson
- with motion modeling: re-mesh with fixed number of mesh nodes